



**Universitat Autònoma
de Barcelona**

**OPEN DEMONSTRATOR
FOR AIR TRAFFIC MANAGEMENT
ANALYSIS**

Memòria del Treball Fi de Grau

en Gestió Aeronàutica

realitzada per:

Roger Ferrer Font

/ dirigida per:

Liana Napalkova

Marko Radanovic

Escola d'Enginyeria

Sabadell, Juliol de 2016

CONTINGUT

Pròleg.....	8
Català.....	8
Castellà.....	8
Anglès.....	9
Agraïments.....	10
Índex d'annexos.....	11
Índex d'il·lustracions i gràfiques	12
Acrònims i abreviatures.....	16
 Introducció	 18
Estat de l'art.....	18
Justificació.....	19
Objectius.....	20
Valor pràctic dels resultats	22
Viabilitat del projecte	23
Novetat del tema.....	25
Membres consorci	25
Estructura de la memòria	27
 1. Operacions de la gestió del tràfic aeri	 29
1.1. Importància del sector aeroportuari.....	29
1.2. Estructura del <i>Air Traffic Management System</i> actual	31
1.2.0. Configuració de l'espai aeri	32
1.2.1. <i>Air Traffic Control</i> : càrrega de treball	33
1.2.2. Resum problemes a resoldre.....	36
1.3. Previsió futura del <i>Air Traffic Management System</i>	37
1.3.0. Nova organització del espai aeri: Single European Sky	38
1.3.1. Plantejament de nous reptes: SESAR	39
1.3.2. Identificació àrees de millora	41

2. Conceptes principals de la proposta	44
2.1. Sistemes de <i>cockpit</i> : TCAS 7.1 vs. E-TCAS	44
2.2. Comunicació dinàmica: <i>multi-agent system</i>	45
2.3. Projecte AGENT: <i>Adaptive self-Governed aerial Ecosystem by Negotiated Traffic</i>	47
2.3.0. Visió i propòsit de creació.....	48
2.3.1. Objectius AGENT aplicats al <i>Air Traffic Management System</i>	50
2.3.2. Resum abast de AGENT	51
 3. Anàlisi i disseny del <i>Open Demonstrator</i>	52
3.1. Anàlisi previ de les condicions.....	52
3.1.1. Metodologia i procediments d'elaboració	52
3.1.2. Finalitat de l'aplicació.....	53
3.1.3. Utilització de les tecnologies de desenvolupament.....	54
3.1.4. Definició dels requisits funcionals principals	56
3.1.5. Representació diagrama de casos d'ús	60
3.2. Disseny conceptual i evolució del <i>Open Demonstrator</i>	62
2.2.1. Visió estàtica dels elements arquitectònics	64
2.2.2. Visió funcional dels elements arquitectònics	65
2.2.3. Visió dinàmica dels elements arquitectònics	75
2.2.4. Interfície gràfica: Vista preliminar a través de <i>mock-ups</i>	91
 4. Cas d'estudi: <i>Stratway – Conflict Detection/Conflict Resolution Tool</i>	97
4.1. Coneixement general de l'algorisme <i>Stratway</i>	97
4.1.1. Definició de la funcionalitat.....	97
4.1.2. Aplicació de les estratègies de resolució de conflictes	98
4.1.3. Informació de sortida	99
4.2. Utilització de la <i>Java Application Programming Interface</i> de <i>Stratway</i>	100
4.2.1. Classes bàsiques	101
4.2.1.1. <i>Stratway.java</i>	101
4.2.1.2. <i>Plan.java</i>	102
4.2.1.3. <i>Detector.java</i>	104
4.3. Seguiment del treball realitzat.....	105
4.3.1. Interfície gràfica: <i>GUI_Stratway.java</i>	109

4.3.1.1. Bounding box search	112
4.3.1.2. Point and distance search	113
4.3.2. Extracció de dades: DataProcessor.java	117
4.3.3. Divisor de waypoints: LegConversor.java	119
4.3.4. Futur treball: IterativeCDCR.java	123
5. Conclusions	127
5.1. Resultats obtinguts	127
5.2. Futur treball	129
6. Referències bibliogràfiques.....	131
6.1. Enllaços web.....	131
6.2. Bibliografia documental.....	132
6.3. Bibliografia escrita	133
7. Annexes i apèndixs	134

PRÒLEG - TREBALL FI DE GRAU DE L'ESCOLA D'ENGINYERIA

Títol del treball Fi de Grau: Open Demonstrator for Air Traffic Management Analysis

Autor: Roger Ferrer Font

Data: Juliol, 2016

Tutors: Liana Napalkova, Marko Radanovic

Titulació: Gestió Aeronàutica

Català: L'objectiu és analitzar i dissenyar una arquitectura d'un ambient de proves (Open Demonstrator) i desenvolupar el mòdul CD / CR utilitzant l'eina Stratway, per provar els conceptes de la propera generació de sistema europeu de ATM. La plataforma permetrà als usuaris finals la incorporació de noves extensions per afegir funcionalitats particulars que demostrin el concepte d'un sistema multi-agent autònom per proporcionar un disseny futur d'un sistema ATM, en el qual els conflictes es preveu amb antelació (tenint en compte les incerteses) que permet diferents resolucions basades en la negociació en el nivell de resolució de conflictes. A la part pràctica de la tesi, s'avaluaran diferents escenaris a través de la informació d'entrada provinent de fonts de dades com FlightStats o DDR2.

Paraules clau: ATM, Detecció i resolució de conflictes, programació, Stratway.

Castellano: El objetivo es analizar y diseñar una arquitectura de un ambiente de pruebas (Open Demonstrator) y desarrollar el modulo CD/CR utilizando la herramienta Stratway, para probar los conceptos de la próxima generación de sistema europeo de ATM. La plataforma permitirá a los usuarios finales la incorporación de nuevas extensions para añadir funcionalidades particulares que demuestran el concepto de un sistema multi-agente autónomo para proporcionar un diseño futuro de

un sistema ATM, en el que los conflictos se prevé con antelación (teniendo en cuenta las incertidumbres) que permite diferentes resoluciones basadas en la negociación en el nivel de resolución de conflictos. En la parte práctica de la tesis, se evaluarán distintos escenarios a través de la información de entrada proveniente de Fuentes de datos como Flightstats o DDR2.

Palabras clave: ATM, Detección y resolución de conflictos, programación, Stratway.

English: The aim is to analyse and to design an architecture of a sandbox (Open Demonstrator) and to develop CD/CR module using Straway tool, for testing the concepts of the next generation of European ATM System. The platform will allow end-users embedding new pluguins to realize particular functionalities demonstrate the concept of an autonomous multi-agent system to provide a future design of an ATM system, in which conflicts are forecasted in advance (considering uncertainties) enabling different negotiation based resolutions at the conflict resolution level. In the practical part of the thesis, different scenarios will be evaluated through the input of data sources such as FlightStats or DDR2 evaluated.

Keywords: ATM, Conflict Detection & Resolution, programming, Stratway.

Agraïments

A la professora i directora de la tesis, Dra. Liana Napalkova, per la seva paciència, coneixement i suport transmès durant aquest període acadèmic.

Al co-director de la tesis, Marko Radanovic, per mantenir un seguiment i col·laboració constant en el projecte.

Al gran company de viatge, Cristian Sánchez, per la seva ajuda en el transcurs de la carrera universitària.

Al Dr. Juan José Ramos, per el suport i ànims a seguir el meu camí.

Al Sr. Ignasi Ingerto i tot l'equip d'Aslogic, per les generoses oportunitats.

I exclusivament,

Dedicat amb tot l'amor possible, a tota la meva família i amics, com a mostra de superació en moments de feblesa

Índex d'annexos (format digital)

Annex 1: Document d'especificacions i requeriments

Annex 2: Exemple de fitxer extret de Flighstats en format (.xlsx)

Annex 3: Exemple de fitxer extret de DDR2 en format (.xlsx)

Annex 4: Arxiu FLSTBarcelona.csv

Annex 5: Arxiu FLSTBruseles.csv

Annex 6: Arxiu FLSTReikiavik.csv

Annex 7: Arxiu FLSTAnkara.csv

Annex 8: Arxiu FLSTHelsinki.csv

Annex 9: Arxiu 20160420_m1.so6

Annex 10: Arxiu 20160423_m1.so6

Annex 11: Arxiu 20160428_m1.so6

Annex 12: Arxiu 20160503_m1.so6

Annex 13: Arxiu 20160511_m1.so6

Annex 14: Arxiu 201603030000_201603312359____EZY_m1.so6

Annex 15: Arxiu 201603030000_201603312359____A320__m1.so6

Annex 16: Arxiu 201603030000_201603312359____B737__m1.so6

Annex 17: Arxiu 201603030000_201603312359____LPL____m1.so6

Annex 18: Arxiu 201604280000_201605260600____IB_m1.so6

Annex 19: Codi font

Índex d'il·lustracions, taules i gràfiques

Figura 1: Esquema descomposició factors de la capacitat de la pista	31
Figura 2: Esquema jurídica dels components dels diferents agents relacionats amb l'ATM Europeu	33
Figura 3: Exemple de l'estructuració del espai aeri del Regne Unit dividit en sectors	33
Figura 4: Gràfic circular en representació del % distribuït entre els factors que afecten en la càrrega de treball del controlador aeri	35
Figura 5: Esquema relacional entre la complexitat d'ATC i la càrrega de treball del controlador	36
Figura 6: Il·lustració de la diferenciació d'espais aeris, amb l'objectiu d'unir-los tots a un únic espai europeu	38
Figura 7: Il·lustració del emblema oficial de SESAR	39
Figura 8: Principals millores en les àrees de capacitat, seguretat, impacte ambiental i eficiència	43
Figura 9: Representació esquemàtica de les diferències principals entre ambdós sistemes	44
Figura 10: Model relacional en representació a les relacions entre els diferents agents i elements	46
Figura 11: Exemple de procés de comunicació entre dos agents	47
Figura 12: Anàlisi de les trajectories alternatives existents davant d'un escenari compost per 4 tràfics diferents	49
Figura 13: logotips oficials respecte seguretat, capacitat i eficiència	50

Figura 14: Emblema oficial del projecte AGENT	51
Figura 15: Model relacional en representació a les interrelacions entre els actors a través dels casos d'ús de l'aplicació	61
Figura 16: Representació esquemàtica de la visió estàtica de l'arquitectura	64
Figura 17: Il·lustració d'algunes components als qual es tindrà accés a través de la GUI layer	65
Figura 18: Representació gràfica del output de la GUI Stratway	66
Figura 19: Captura de pantalla de la interfície de CPN Tools	67
Figura 20: Captura de pantalla de la interfície de NetLogo	69
Figura 21: Representació dels components que inclou la Data layer	69
Figura 22: Captura de la pantalla d'inici de la font de dades Flightstats	70
Figura 23: Captura de la gràfica d'arxius descargables a través de DDR2	71
Figura 24: Representació dels components que inclou la module layer	71
Figura 25: Diagrama de flux d'un exemple d'algorisme destinat a la detecció i resolució de conflictes	72
Figura 26: Graf d'exemple en l'anàlisi de diferents escenaris	73
Figura 27: Esquema il·lustratiu d'una comunicació i connexió entre diferents elements	74
Figura 28: Il·lustració del funcionament del sistema TCAS	74
Figura 29: Representació gràfica de possible indicadors KPI	75
Figura 30: Esquema de la interacció entre els diferents agents involucrats	76
Figura 31: Diagrama procedimental de les relacions entre els diferents casos d'ús i accions	77
Figura 32: Diagrama de pantalles que mantenen una relació entre elles	78

Figura 33: Esquema de seguiment de representació de les opcions d'acció	79
Figura 34: Il·lustració animada del funcionament i estructura aplicació	80
Figura 35: Representació gràfica de les interrelacions existents entre les diferents funcionalitats establertes per mòduls i capes	82
Figura 36: Representació gràfica final dels components de l'aplicació	84
Figura 37: Representació de la capa de la interfície gràfica d'usuari general	85
Figura 38: Representació de la capa de les interfícies distribuïdes	85
Figura 39: Representació de l'última capa modular (back-end)	86
Figura 40: símbol primer paquet modular	87
Figura 41: símbol segon paquet modular	88
Figura 42: símbol tercer paquet modular	89
Figura 43: símbol quart paquet modular	90
Figura 44: símbol cinquè paquet modular	90
Figura 45: Primera captura de pantalla interfície gràfica	92
Figura 46: Segona captura de pantalla interfície gràfica	93
Figura 47: Tercera captura de pantalla interfície gràfica	94
Figura 48: Quarta captura de pantalla interfície gràfica	94
Figura 49: Cinquena captura de pantalla interfície gràfica	95
Figura 50: Sisena captura de pantalla interfície gràfica	96
Figura 51: Exemple arxiu csv	98
Figura 52: Representació estratègia Track	99
Figura 53: Representació estratègia Vertical	99

Figura 54: Representació estratègia Velocity	99
Figura 55: Representació estratègia SideStep	99
Figura 56: Il·lustració de tots els paquets i llibreries	105
Figura 57: Il·lustració dels paquets i classes principals	106
Figura 58: Il·lustració de les llibreries utilitzades	107
Figura 59: Il·lustració de les classes pròpies de Stratway	108
Figura 60: Captura de pantalla de la GUI Stratway	109
Figura 61: interfície gràfica valors paràmetres	110
Figura 62: Exemple d'arxiu en format JSON	111
Figura 63: exemple de cerca a través de bounding box	112
Figura 64: exemple de cerca a través de point and distance	113
Figura 65: exemple arxiu format .s06	114
Figura 66: interfície gràfica de filtre intern	116
Figura 67: interfície gràfica de introducció de dades	116
Figura 68: interfície gràfica de LegConversor	120
Figura 69: comparació entre arxius csv: arxiu anterior	120
Figura 70: comparació entre arxius csv: arxiu posterior	121
Figura 71: Finestra de càrrega d'arxius	122
Figura 72: Captura de pantalla amb trajectòries GUI_Stratway	122
Figura 73: Panell d'informació sense solucions	124
Figura 74: Panell d'informació amb resolució	126

Acrònims i abreviacions

CDM: Collaborative Decision Making	ATS: Air traffic services
A-CDM: Airport Collaborative Decisión Making	FIS: Flight Information Services
AENA: Aeropuertos Españoles y Navegación Aérea.	NM: Network Manager
E-TCAS: Enhanced-Traffic Collision Avoidance System	RBT: Reference Business Trajectories
STREAM: Strategic Trajectory de-confliction to Enable seamless Aircraft conflict Management	CDM: Colaborative Decision Making
SID: Standard Instrument Departures	DMT: Decision Making Tools
SWIM: System Wide Information Managment	AU: Airspace Users
SESAR: Single European Sky ATM Research	I+R: Investigation + Research
TCAS: Traffic Collision Avoidance System	EU: European Union
CNS: Communication Navigation Surveillance	UML: Unified Modeling Language
ANS: Air navigation services	IDE: Integrated Development Environment
ATM: Air Traffic Management	COBOL: Common Business- Oriented Language
ATC: Air Traffic Control	HTML: Hyper Text Mark-up Language
ATCO: Air Traffic Control Operator	GUI: Graphical User Interface
ASM: Air Space Management	CSV: Comma Separated Values

ATFMU: Air Traffic Flow Management Unit

CFMU: Central Flow Management Unit

KPI: Key Performance Indicator

AGENT: Adaptive self-Governed aerial
Ecosystem by Negotiated Traffic

RBT: Reference Business Trajectory

BT: Business Trajectory

ATFM: Air Traffic Flow Management

PIB: Preu Interior Brut

FL: Flight Level

FAB: Functional Airspace Block

TBO: Trajectory Based Operations

CTA: Controlled Time of Arrival

ADS-B: Automatic Dependent Surveillance
Broadcast

TAT: Target Arrival Time

CTOT: Calculated Take-off Times

DDR2: Data Demand Repository 2

CD&CR: Conflict Detection & Conflict
Resolution.

XML: Extensible Markup Language

CSS: Cascade Style Sheets

API: Applitacion Programming Interface

JSON: Javascript Object Notation

FSC: Flights scheduling and coordination

ACAS: Aircraft Collision Avoidance System

SES: Single European Sky

LANTAK: Landing and Take-Off Scheduling
Platform.

RPA: Remote Pilot Aircraft

TA: Traffic Advisories

RA: Resolution advisories

CA: Conflict Avoidance

INTRODUCCIÓ

Estat de l'art

Actualment les tecnologies de la informació permeten una gran varietat d'aplicació en diferents camps, i un d'ells, és el sector aeronàutic. L'espai aeri europeu es caracteritza per una **estructura molt rígida de la xarxa de rutes àrees**. La resolució dels conflictes entre vols, a causa de les limitacions de l'espai aeri, incrementa la càrrega de treball del ATC i restringeix el flux de trànsit.

Gràcies a la **creació de nous sistemes de detecció o de comunicació** s'han pogut aconseguir millorar molts dels processos que actualment permeten que els vols tinguin una seguretat garantida, i per tant, ajuden a prendre les millors decisions destinades a la recerca de la màxima eficiència.

Un dels nous reptes que es preveuen referents a la gestió del tràfic aeri, com ja s'ha dit prèviament, és buscar noves solucions davant de l'augment del volum d'aquest tràfic. S'han discutit noves perspectives sobre la gestió i control del ATM en diferents conferències i s'han proposat noves maneres d'automatització i de desenvolupament sistemàtic, considerant diversos enfocaments relacionats amb la **substitució de l'actuació humana per la tecnologia**.

Existeix una gran varietat de projectes de recerca i desenvolupament que comparteixen el mateix àmbit, enfocats en l'aportació de noves solucions de problemes actuals. Alguns dels exemples més rellevants són els següents:

STREAM (*Strategic Trajectory de-confliction to Enable seamless Aircraft conflict Management*): L'objectiu principal d'aquest projecte és la investigació en la

detecció i resolució de conflictes d'una forma, computacional, eficient per mitjà d'algoritmes de resolució estratègica de conflicte, amb un marge de pocs segons o minuts, tenint en compte les preferències i les restriccions de la xarxa.

ALICIA: L'objectiu principal és el de millorar l'eficiència de temps en el sistema de transport aeri mitjançant el desenvolupament de nous sistemes de cabina que permetin un nombre significativament major de moviments d'aeronaus del que és possible avui dia. D'aquesta manera es busca aconseguir operar en gairebé totes les condicions meteorològiques i volar més a prop a un risc més baix.

Aquests entre d'altres, són molts dels projectes que han proposat **canvis innovadors** en diferents àmbits donant rellevància en l'actuació de les noves tecnologies.

Concretament aquest treball present esta enfocat bàsicament en una proposta basada en aquesta idea. Amb el suport de diferents institucions internacionals, es busca una tendència a **millorar la manera de treballar que tenen els actuals usuaris del espai aeri**.

Justificació

Cada vegada tot canvia més ràpid degut al gran creixement poblacional i al continu desenvolupament de noves tecnologies. Actualment el sector aeronàutic esta vivint una sèrie de canvis constants corresponents a diferents àmbits, però aquesta tesi es centra en l'estat del pilar operacional del avions: **La gestió del tràfic aeri (ATM)**.

Es preveu un creixement del trànsit aeri, d'entre un 40% i 70% durant els propers 8-14 anys. Aquest creixement exponencial de la capacitat de la xarxa de transport europea es pot convertir en una situació problemàtica si no es

resol a temps (tan en temes econòmics, socials o fins i tot ambientals). Partint d'aquestes previsions exposades, es prioritza la **recerca de noves solucions** amb la finalitat d'aconseguir satisfer aquesta **demanda de capacitat, eficiència i seguretat**.

Davant d'una falta de capacitat en les pistes, la gestió d'aquest espai aeri es converteix en un dels agents primordials a adaptar a les previsions de tràfic. Aquesta gestió de la capacitat de l'espai aeri esta condicionada per dos factors:

- La possible i limitada **separació entre els diferents vols** existents en l'espai aeri.
- La **càrrega de treball del controlador** del tràfic aeri.

Així que contemplant aquests dos principals reptes, és necessari treballar en la recerca de possibles solucions que ajudin a superar-los. Tenint en compte que el primer problema descrit es tracta d'un inconvenient que en mesura, sempre hi serà com a límit espacial (tot i que es pugui ajustar i buscar la màxima optimització de les trajectòries), l'elaboració del present treball s'ha destinat a focalitzar-se en el segon problema (sense oblidar el primer), ja que aquest permet la utilització d'una àmplia varietat d'eines per solucionar-lo.

Objectius

Una vegada s'ha deixat clar el motius que han comportat l'acció de desenvolupar aquest tema del projecte, és necessari marcar els objectius que es volen assolir. L'objectiu principal del projecte és **participar en l'anàlisi i desenvolupament d'un ambient de proves (Open Demonstrator) per provar els conceptes de la propera generació de sistema europeu de ATM**.

Objectius generals:

Per aconseguir aquest objectiu principal, s'han formulat els següents objectius referents al desenvolupament del projecte:

1. Adquisició d'un nivell de **coneixement general** de manera que permeti realitzar un anàlisi concís de l'estat del sistema ATM actual, per tal d'identificar les deficiències i els punts febles d'aquest, en els quals l'aplicació pràctica del projecte puguin millorar-ne l'eficiència i el rendiment.
2. Analitzar l'**aplicabilitat de l'aplicació**, detallant les funcionalitats específiques de cadascun dels seus mòduls, quin tipus de tractament de la informació es realitza, quines són les diferents fluctuacions de dades que es comparteixen a través de les connexions i interrelacions, etc. Tot per arribar a entendre el conjunt de les diferents eines incorporades al sistema amb la intenció de resoldre diversos aspectes de la gestió del tràfic aeri.
3. En base a l'estructura prèviament exposada, desenvolupar l'eina principal del Open Demonstrator i fase inicial de tractament de les dades introduïdes (anomenada *Stratway*) que s'encarrega de parametritzar la informació de les trajectòries i aplicar un **algorisme destinat a detectar i resoldre els possibles conflictes** entre les diferents trajectòries d'una forma autònoma i automàtica, i per tant, reduir les fonts de sobrecàrrega de treball dels controladors aeris.
4. Finalment exposar en format escrit un **resum conclusiu** que englobi parts sintetitzades de la recerca, anàlisi, disseny i desenvolupament, presentant una evolució i resultat de tot el procés d'elaboració del treball.

Objectius concrets de l'aplicació:

Posant l'objectiu descrit número 3 com a subjecte, és a dir, el desenvolupament del cas d'estudi, es poden especificar una sèrie d'objectius més específics i tècnics relacionats amb l'aplicació pràctica de l'eina *Stratway*:

1. Explicació de l'objectiu, l'aplicabilitat de l'eina i dels paquets que engloba.
2. Addició de funcions relacionades amb l'extracció i manipulació de dades.
3. Permetre noves funcionalitats, com la d'assignació iterativa de vols a través d'un cicle continu, entre d'altres possibles, exposades per mitjà d'exemples pràctics.

Valor pràctic dels resultats

Aquesta plataforma està destinada a permetre provar, crear i validar diferents tecnologies i mètodes que vagin destinats a la millora de la gestió del tràfic aerí, per aquest motiu es pot afirmar que es tracta d'un projecte amb un valor pràctic **considerablement alt**. També la recerca i l'anàlisi de tota la informació que envolta aquest tema per tal d'entendre l'abast que pot arribar a tenir el projecte té una gran importància. Els **principals beneficis** d'utilitzar aquesta eina són:

- Reducció de la càrrega de treball dels controladors.
- Optimització del temps de resolució de conflictes de les trajectòries.
- Fàcil i múltiple comunicació entre els diferents agents involucrats en un possible conflicte.
- Capacitat de resolució directa, amb una reducció d'intervenció d'ATC.
- Adaptabilitat de diferents tipus de dades i d'eines de validació en un mateix conjunt.
- Simulació i visualització gràfica de les dades processades.

- Compatibilitat amb altres sistemes com el TCAS per millorar la seguretat.
- Possibles futurs estalvis econòmics i impactes ambientals positius.

Viabilitat del projecte

Durant l'elaboració del treball s'estudien diferents casos des de diferents punts de vista i com ja s'ha dit, el objectiu més important del treball és participar en l'anàlisi i desenvolupament d'un ambient de proves per tal de facilitar la re-configuració del nou ATM a les futures necessitats.

Per a dur a terme el treball, s'ha hagut de realitzar un **filtre de la informació molt exhaustiu**, ja que al tractar-se d'un tema que es localitza en la fase d'inicialització i de desenvolupament del seu conjunt, calia establir les bases intentant destriar la informació més rellevant d'entre les poques fonts de documentació existents. Bàsicament, la informació cercada i les conclusions extretes són **d'interès personal** i per a tothom qui vulgui tenir una **referència de la situació actual**, ja que això els hi facilitarà una ampliació dels coneixements generals sobre el sector aeronàutic i concretament en el paper intern de l'ATM europeu.

El interès del recent i inicial disseny d'un Open Demonstrator podria anar sobretot dirigit exclusivament a empreses del sector aeronàutic, al tractar-se d'un projecte internacional on diferents institucions europees hi estan participant, com EUROCONTROL a través del projecte SESAR, i amb aquest treball es permetrà entendre una prèvia idea d'aquest canvi que s'espera desenvolupar. Una vegada finalitzat i validat el projecte, les aportacions i beneficis principals de la seva posada en funcionament estarien directament relacionats amb la seguretat, la capacitat i l'eficiència de les operacions d'ATM.

Alhora d'elaborar el projecte és necessita tenir un coneixement general del **panorama en el que es veu involucrat**. També és necessari el coneixement del conceptes que fan referència les **tècniques bàsiques de disseny**. És imprescindible estar familiaritzar amb els **temes de la gestió aeronàutica** actual, quins són les eines més usades i quina aplicació tenen. Al tractar-se d'un projecte d'empresa i amb un alt valor pràctic, cal tenir el **domini en la utilització de les tecnologies descrites** en punts definits més endavant.

No es descarta l'**existència de riscos** i possibles situacions adverses a la finalització adequada del producte, uns exemples serien una **mala gestió del temps** i de seguiment de les tasques a realitzar, i que això provoqués una modificació període esperat de termini. **La falta d'informació** prèviament tractada sobre un tema en fase d'investigació com és aquest, pot comportar una carència d'enteniment del projecte i pot crear una percepció difusa del usuari respecte la idea global.

Per l'altra banda, també es pot destacar els possibles **impactes positius** a nivell personal, que es poden esperar del treball. Aquest treball pot aportar una visió molt àmplia dels canvis i temes més importants que s'estan estudiant actualment en el sector aeronàutic. També permetrà **aplicar i practicar molts coneixements adquirits** durant el transcurs del grau, tant teòrics com tècnics, en un projecte del món real suportat per diferents empreses i institucions oficials. El treball comportarà unes millores en les **capacitats organitzatives i relacionals** de la persona, ja que es tracta d'un projecte comú on les persones involucrades tenen perfils diferents. Al ser realitzat basat en un projecte d'empresa, a nivell professional, permetrà **nous aprenentatges i maneres de treballar** en experiències prèviament desconegudes obrint-se així, a nous horitzons laborals.

Novetat del tema

La novetat d'aquesta tesi resideix principalment en el disseny conceptual d'un projecte que, encara que estigui en una fase de laboratori, té l'objectiu de resoldre **problemes existents i reals**, dispersos en la xarxa d'aeroports europeus.

La seva aplicació al sistema de control aeri implicarà, d'una forma considerablement dràstica, un canvi en la manera general de gestionar el tràfic aeri. Existeixen diferents eines individuals que permeten l'anàlisi de dades d'una forma eficient i ràpida, però no existeix **cap eina que permeti l'adaptabilitat de diferents eines** en un mateix conjunt i que aquestes actuïn d'una forma paral·lela i cooperativa.

El procés d'elaboració dels casos d'estudi s'ha basat en el processament de dades real, proveïdes per fonts veritables de **trajectòries a temps real**. Aquests casos d'estudi s'han desenvolupat i provat amb plataformes de software certificades i oficials.

Membres consorci

Com a treball realitzat en l'empresa, en aquest apartat es detalla la informació general de les organitzacions i institucions acadèmiques que formen part del consorci del projecte, compartint informació i participant-hi directament.

Universitat Autònoma de Barcelona, com a coordinadora del projecte, és una universitat pública catalana creada el 1968. La major part dels seus centres docents i serveis són al Campus de Bellaterra, situat a Cerdanyola del Vallès (Vallès Occidental). Té també centres docents a Sabadell (Vallès Occidental), Manresa (Bages) i a la ciutat de Barcelona. La UAB acull 226 grups de

recerca, 57 departaments, 45 instituts i centres, així com 5 hospitals, que tenen una intensa activitat investigadora, reconeguda internacionalment.

Aslogic és una organització que participa directament amb la Universitat Autònoma de Barcelona (UAB). La seva missió és generar solucions en la presa de decisions per a les organitzacions i els ciutadans, apostant per el coneixement d'alt valor. Les solucions d'Aslogic han estat desenvolupades i validades per prestigioses organitzacions nacionals i internacionals. El programa de Responsabilitat Social que segueix es basa en valorar la inversió pública en recerca, participació en projecte internacionals i innovadors, facilitació de la inserció laboral a recent graduats, democratització en el disseny de les polítiques d'actuació, respecte amb el medi ambient, entre d'altres.

El present document, en representació de la participació en el projecte tractat, s'ha realitzat amb el suport i sota la supervisió dels principals membres que representen al projecte d'aquestes dues organitzacions anterior.

També és important esmentar les dues institucions participants internacionals que són:

Cranfield University és una universitat de postgrau i basada en la investigació britànica especialitzada en la ciència, l'enginyeria, la tecnologia i la gestió. Esta estructurada en dos campus: el campus principal es troba a Cranfield, Bedfordshire, i el secundari, és l'Acadèmia de Defensa del Regne Unit a Shrivenham. El campus principal és únic al Regne Unit per tenir un aeroport operacional al costat d'ell. Les instal·lacions de l'aeroport són utilitzats pels propis avions de la Universitat de Cranfield durant el curs de la recerca i la investigació aeroespacial.

TU Braunschweig és la universitat tècnica més antiga d'Alemanya. Va ser fundada el 1745 com Collegium Carolinum i és membre de TU9, una societat

constituïda dels més famosos i grans instituts alemanys de la tecnologia. Està classificada entre les millors universitats d'enginyeria a Alemanya. Avui compta amb quasi bé 19.000 estudiants, pel que és la universitat més gran de Saxònia després de la Universitat de Göttingen i la Universitat de Hannover.

Estructura de la memòria

Referent a l'estructura d'aquesta memòria, la informació exposada està organitzada de manera que es detallin i es completin els següents punts:

Punt 1: En aquesta part es realitza una descripció general, bàsicament teòrica, del context en el que es troba el objecte estudiat i de diferents conceptes aeronàutics (que s'han assolit en els transcurs del grau) que cal conèixer degut a la seva afectació, per tal de reflectir la situació real d'una manera àmplia.

Punt 2: En aquest punt es comença a introduir-se en el cos del projecte. Es basa en una descripció i representació del procés de disseny de l'arquitectura de tot el projecte (Open Demonstrator). També s'hi adjunten alguns punts referents al document d'especificacions de requeriments i una primera imatge gràfica del projecte global.

Punt 3: En aquest punt s'hi exposa la part pràctica realitzada amb la eina principal del cas estudiat: Stratway. Aquesta demostració és portada a terme per mitjà de casos d'exemple sobre les funcionalitats adherides, complementada amb gràfiques i il·lustracions del funcionament.

Punt 4: Finalment, se'n extreuen unes conclusions de tot el procés de treball en les que s'hi reflecteixin una evolució i un aprenentatge dels aspectes més rellevants, conjuntament amb una opinió ben fomentada de l'experiència.

Punt 5: Apartat on es llisten tota la sèrie de cites bibliogràfiques, fonts d'on he extret informació, gràfiques, figures o qualsevol altre recurs que hagi utilitzat en l'elaboració d'aquesta memòria.

Punt 6: Referència als annexos i apèndixs principals, diferents documents externs al cos del projecte, però relacionats amb l'elaboració de les seves parts constituents.

1. OPERACIONS DE LA GESTIÓ DEL TRÀFIC AERI

Per mirar d'obtenir una prèvia idea teòrica dels temes tractats durant l'elaboració d'aquesta tesis, és necessari tenir el coneixement de diferents conceptes que permeten construir una visió clara del panorama d'actuació principal (ATM). En aquest punt es pretén il·lustrar i exposar d'una forma més explicativa i més extensa, alguns dels temes que s'han citat en la part introductòria o que es citaran en l'elaboració de la resta de parts del treball, per tal de poder entendre amb més exactitud la importància de la gestió del tràfic aeri i comprendre el paper del present projecte en aquest marc.

1.1. Importància del sector aeroportuari

Els aeroports europeus són àmpliament reconeguts per tenir un **impacte econòmic i social considerable** en les seves regions circumdants.

La **globalització** de l'economia mundial és un factor clau de creixement del trànsit aeri. **L'activitat econòmica** a tot el món és el factor propulsor més potent del creixement del transport aeri comercial i la consegüent demanda d'avions. Un **augment de la població urbana** ha portat històricament a un augment del PIB per càpita, un factor clau per a l'aviació. Això crea la necessitat de viatjar per **relacions entre clients, proveïdors i diferents parts** d'una empresa.

La restricció de la capacitat de l'aeroport o la gestió de la demanda de vols podria tenir **conseqüències dràsticament negatives**. El fet de no augmentar la capacitat per satisfer la demanda podria reduir el PIB a nivell nacional o regional en un 2,5 a un 3%. Les restriccions en el creixement dels vols podrien causar:

- Tarifes més altes
- Menys passatgers
- Increment dels costos indirectes a les empreses
- Reducció del serveis aeris
- Accés restringit de les comunitats remotes

Capacitat de les pistes

La capacitat de les pistes és la capacitat de processament d'una pista sobre un cert període. Quan la demanda de servei de la pista és més gran que la capacitat de producció, les aeronaus que surten han d'esperar a la cua de sortida i les aeronaus que arriben han d'esperar a la pica de la celebració d'espai aeri. **L'objectiu és maximitzar el rendiment de la pista.**

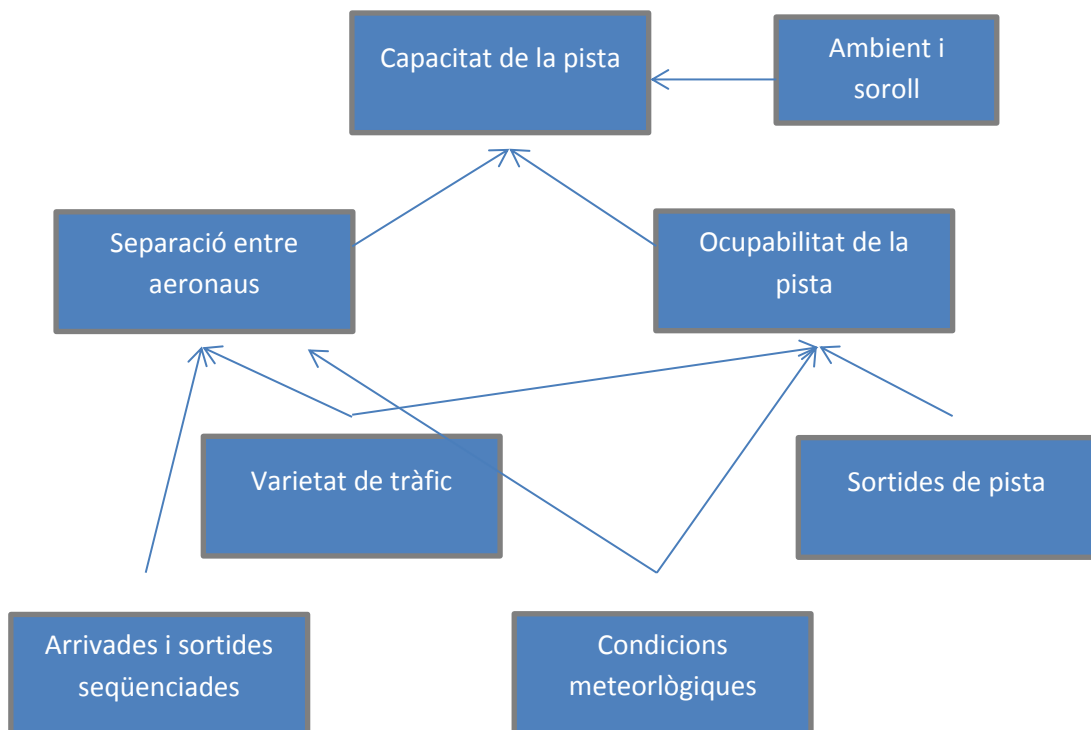


Figura 1: Esquema descomposició factors de la capacitat de la pista

Com aconseguir aquest objectiu?

- **Reduir les separacions** entre arribades durant la fase d'aproximació final de vol.
- Estandarditzar les velocitats terminals
- **Reduir els temps d'ocupació** de la pista i els temps de resposta del pilot
- **Millorar la comunicació** entre els socis aeroport

Quins factors hi influeixen?

- Els criteris de separació espacial-geomètriques
- càrrega de treball del controlador de trànsit aeri

1.2. Estructura del *Air Traffic Management System* actual

ATM és un subsistema del ANS que es compon de persones, procediments i equips que proporcionen ATM i serveis de CNS. ATM esta compost d'una **banda terra i una banda aire**, totes dues parts són necessàries per garantir un moviment segur i eficient de les aeronaus en totes les fases del vol.

El **ATM Europeu** és el resultat de la integració de diversos sistemes, processos i recursos, i els principals i que inclou són:

- Gestió de l'espai aeri (ASM).
- Gestió del flux del tràfic aeri (ATFM).
- Control del tràfic aeri (ATC).

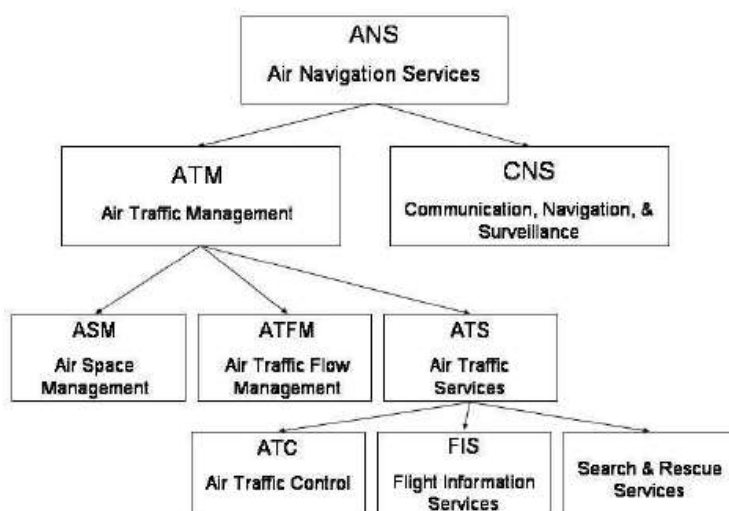


Figura 2: Esquema jurídica dels components dels diferents agents relacionats amb l'ATM Europeu

Pel que fa a la capacitat de l'ATM, és necessari assenyalar la **situació actual**. En últimes dècades, el sistema de transport aeri ha experimentat **un important creixement de la demanda**, probablement a causa de factors:

- **Noves relacions intercontinentals** (causa directa de la territorial, polític i econòmic unificació del continent).
- **Augment comoditat, rapidesa i seguretat** que el transport aeri.
- Considerable caiguda dels preus en el mercat des de la **introducció de les companyies aèries de baix cost**.

1.2.0. Configuració de l'espai aeri

l'espai aeri regulat es divideix en volums d'espai anomenats **sectors**, i està subjecte a regles de navegació internacionals que dicten com se'ls permet aeronaus per operar. L'espai aeri europeu esta actualment dividit en una complexa xarxa de sectors gestionats per **41 centres de control** de trànsit aeri.

El sistema de les **vies respiratòries** també esta fixada, i pot ser pensat com l'equivalent a la xarxa d'autopistes de diversos països: poden tenir dos carrils, un carril per al vol dels avions d'est a oest i un carril per als avions que

viatgen d'oest a cada un. Aquests carrils també poden tenir un nombre de **diferents pisos o nivells**, per exemple, trànsit del est a oest pot volar en el nivell de vol FL250 (és a dir, 25.000 peus) i d'oest a est en el tràfic FL260. Tal organització implica que l'FL250 i FL260 hi haurà un nombre de carrils de trànsit dels avions que volen tots en la mateixa direcció.

L'espai aeri europeu es caracteritza per una estructura molt rígida de la xarxa de routes àrees. La resolució dels conflictes habituals, a causa de les limitacions de l'espai aeri, **incrementa la càrrega de treball del ATC i restringeix el flux de trànsit.**

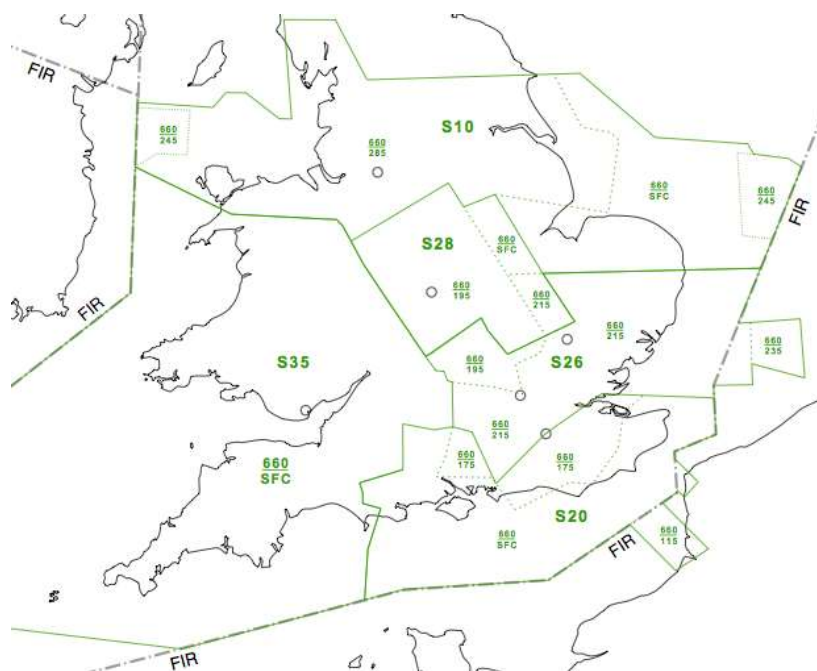


Figura 3: Exemple de l'estructuració del espai aeri del Regne Unit dividit en sectors

1.2.1. Air Traffic Control: càrrega de treball

La **seguretat és el primer objectiu de l'ATM** i per tant també l'objectiu principal de l'obra d'un controlador. Depenent de la fase de vol, avions són controlats per la torre de control d'aeròdrom, la unitat de control d'aproximació o centre de control d'àrea. Independentment de la fase de les operacions de vol **la funció principal d'ATC es pot dividir en quatre parts:**

- entrada d'aeronaus en l'àrea de responsabilitat;
- determinar les futures posicions de les aeronaus i els possibles conflictes entre ells;
- separació de les aeronaus;
- sortida de les aeronaus des de l'àrea de responsabilitat.

Control de l'evolució d'un vol:

- Els aeroports de molt trànsit tenen rutes de sortida i arribada inicials estandarditzades.
- Una vegada que l'avió s'enlaira el pilot rep instruccions per posar-se en contacte amb el control de la terminal freqüència. El controlador d'àrea terminal li donarà al pilot informació sobre com continuar el ascens.
- El vol progressarà amb una sèrie d'instruccions sobre l'altitud, rumb i velocitat, així com canvis de freqüència que corresponen a la comunicació amb els diferents controladors en ruta, de diferents sectors.
- Al apropar-se a un aeroport, el controlador en ruta transferirà la responsabilitat de l'aeronau a la zona de la terminal, des del punt de vista del pilot, això correspon a un canvi de freqüència.
- El controlador de terminal proporcionarà instruccions correctes que ha de seguir: la ruta de descens, així com les condicions meteorològiques.
- En aterrar, se li demanarà al pilot posar-se en contacte amb la freqüència del controlador de torre, que guiarà l'aeronau al seu stand.

La feina d'un controlador de trànsit aeri es pot descriure com una persona encarregada de **prendre decisions amb un objectiu, que es realitzen en un entorn dinàmic i canviant, i a temps real, i amb una dependència constant d'altres persones i de la tecnologia disponible.** Això provoca un gran estrès ja requereix prendre les decisions adequades en el moment adequat.

Per tant un controlador es refereix sovint com a **gestor d'un sector de l'espai aeri**, que ha de trobar l'equilibri entre les demandes de trànsit i el nivell de càrrega de treball, per mirar de detectar i resoldre conflictes entre els diferents vols.

La **capacitat** d'un sector de Control de Trànsit Aeri (ATC) és el **nombre màxim d'aeronaus que poden entrar en un sector en un període de temps determinat**, permetent al mateix temps un nivell acceptable de la càrrega de treball del controlador. Encara que la capacitat en ruta de l'espai aeri pot determinar-se a partir limitació espacial geomètrica, a mesura que augmenta el trànsit aeri, la **càrrega de treball del controlador és el factor principal**.

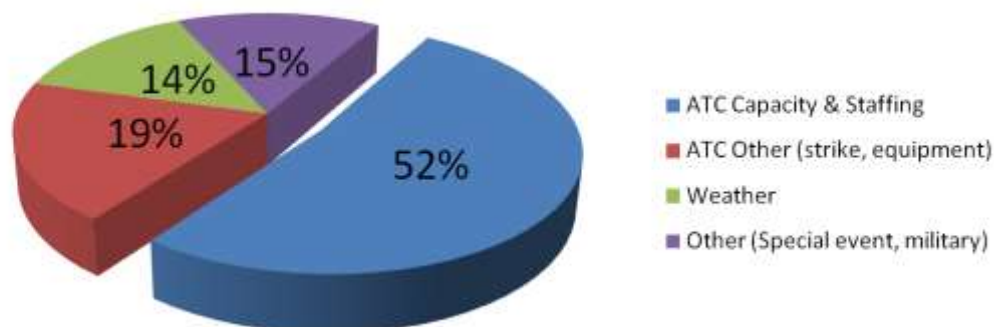


Figura 4: Gràfic circular en representació del % distribuït entre els factors que afecten en la càrrega de treball del controlador aeri

La càrrega de treball experimentada pels controladors de trànsit aeri es veu afectada per la complexa interacció entre:

- La **situació en l'espai aeri** (les característiques tant del trànsit aeri i el sector).
- L'**estat dels equips** (la fiabilitat i la precisió dels equips de la sala de control i en l'aeronau).
- L'**estat del controlador** (l'edat, l'experiència, les estratègies de presa de decisions).

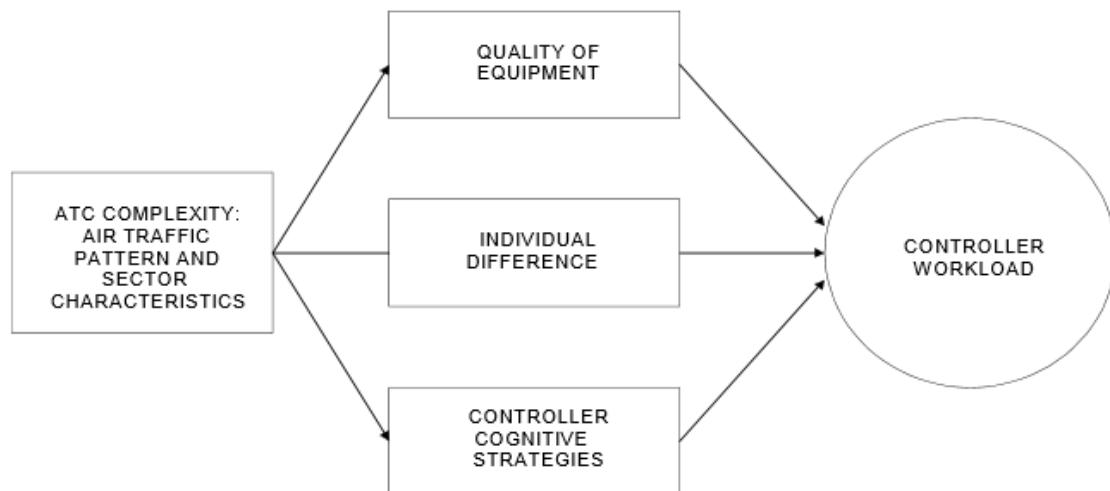


Figura 5: Esquema relacional entre la complexitat d'ATC i la càrrega de treball del controlador

L'única manera d'augmentar la capacitat de l'espai aeri i reduir la càrrega de treball ATCO és per mitjà de **delegació d'algunes de les tasques a altres serveis o sistemes automatitzats**.

1.2.2. Resum problemes a resoldre

Un cop s'ha exposat el estat del ATM actual, cal identificar quins són les principals **deficiències i debilitats** en el seu funcionament, i conseqüentment trobar les possibles àrees de millora:

- **La falta de integració entre ATCFM i ATC:** Aquesta manca causa pobres prediccions de trànsit i alta incertesa i desincronització sobre el trànsit, augmentant així el nombre i la durada dels retards en els vols, així com la càrrega de treball de l'ATC.
- **La manca de flexibilitat per a una millor planificació tant de la espai aeri i els vols durant la fase d'execució:** La manca de mecanismes adequats per a la coordinació entre ATCFM, l'ATC i els AU impedeix la planificació dinàmica

eficient en temps real en el qual es refereix a les rutes de l'espai aeri i a l'optimització dels recursos.

- **Poca automatització d'alguns procediments d'ATM i escassetat de noves tecnologies:** La majoria de les prediccions a curt termini i mitjà termini són fetes pels controladors i els gestors de moviments mirant pantalles de trànsit aeri i extrapolant mental la situació, per tant amb poca ajuda d'automatització.
- **La manca d'un marc model de trànsit comú compatible i global per integrar tecnologies diferents i modernes:** el corrent ATM implica múltiples prenedors de decisions que avui en dia no poden coordinar-se adequadament entre els diferents sectors.
- **Manca d'unificació de formació de personal,** causant moltes diferències d'equip i capacitat, donant lloc a una situació on els processos no sempre estan encaminats cap a una mateixa direcció. Sense esmentar, que diferents països europeus presenten diferents reglaments de la circulació aèria.

1.3. Previsió futura del *Air Traffic Management System*

Les previsions de creixement de trànsit aeri realitzades des del 1989 s'han incrementat en un 33% i s'espera que gairebé es dupliqui en els propers 20 anys. A causa de la forta expansió en el tràfic aeri, s'ha esdevingut a una situació amb **manca de fluïdesa en el trànsit aeri**. L'obtenció de la capacitat màxima possible amb les tècniques, mètodes i procediments actuals de la gestió del d l'espai aeri europeu, ja s'ha assolit i, per tant, és **necessària la introducció de nous conceptes** en el futur camp de la gestió del trànsit aeri siguin operativament aplicables per tal **d'aconseguir un escenari ATM capaç de gestionar el trànsit aeri previsible**.

1.3.0 Nova organització de l'espai aeri: Single European Sky (SES)

La **Comissió Europea** va decidir iniciar el procés d'eliminació de les fronteres de l'espai aeri, amb la intenció d'**organitzar i gestionar l'espai aeri europeu i la navegació aèria d'una manera aliada**, concebant així el que es coneix com el **cel únic europeu**. Aquest procés, que encara és viu i no s'espera per posar fi a fins almenys 2030, portarà a la modernització total de les tecnologies i els procediments utilitzats per gestionar el trànsit aeri a Europa.

Prèviament **EUROCONTROL** té la necessitat de dissenyar el futur ATM europeu, per tal d'adaptar-lo a les noves necessitats i oportunitats del cel únic europeu.



Figura 6: Il·lustració de la diferenciació d'espais aeris, amb l'objectiu d'unir-los tots a un únic espai europeu.

Per tant, el desenvolupament del projecte d'un únic cel europeu busca aconseguir diferents objectius relacionats amb la **seguretat, la capacitat, la eficiència i l'impacte ambiental**:

- millores rellevants en la capacitat del espai aeri.
- Reducció dels costos i l'augment de de l'eficiència del sistema mitjançant la millora de les rutes aèries disponible entre els aeroports i les maniobres i procediments executats.
- Minimització de l'impacte ambiental negatiu del sistema de transport aeri.

1.3.1. Plantejament de nous reptes: SESAR

Moltes de les tecnologies actuals, es van implantar fa anys, i responen a les necessitats passades de l'espai aeri, però no a les necessitats actuals. Això a provocat que algunes de les decisions dels controladors siguin ineficients a causa de d'aquestes eines de suport antigues i procediments de gestió obsolets, que actualment estan establerts.

El programa SESAR té l'objectiu de fomentar una **modernització completa de les tecnologies** que donen suport a la futura ATM europeu, juntament amb EUROCONTROL per mirar **d'unir i coordinar els grups** més rellevants de la Unió Europea, que poden estar interessats en dedicar esforços a la investigació i desenvolupament, per tal de fer que el cel únic europeu entri en una realitat.

Tres fases es planifiquen per a l'execució inicial del programa SESAR:

- **Fase de definició** (2005-2008).
- **Fase de desenvolupament** (2008-2016).
- **Fase d'aplicació** (2014-2020).



Figura 7: Il·lustració del emblema oficial de SESAR.

Les bases d'aquest futur ATM, es regeixen per una sèrie de nous sistemes i eines de l'ATM que proposa **SESAR**:

Business Trajectory (4D trajectories): La trajectòria de l'aeronau prevista s'anomena *Business Trajectory* (BT). Aquesta trajectòria es pot planificar en diferents fases: **BDT**, **SBT** i **RBT**. El procés de planificació de la BT comença quan l'usuari de l'espai aeri crea el pla de vol desitjat i acaba quan el vol ha arribat amb èxit a l'aeroport de destinació. Si aquest sistema s'acaba utilitzant, les trajectòries hauran de ser definides amb molta precisió com a 4D trajectòries (3D + temps) i executades amb precisió, per tal de resoldre tots els conflictes tenint en compte les condicions climàtiques.

Colaborative Decision Making (CDM): La presa de decisions en col·laboració permetrà arribar a una solució que satisfaci a totes les parts interessades, per tal de millorar l'eficiència i optimització de diferents factors. Per assegurar que aquest sistema funciona com es preveu, totes les parts interessades haurà de tenir l'accés a tota la informació rellevant.

Integració d'operacions aeroportuàries: La plena integració dels aeroports en el procés de planificació de la capacitat europea és essencial per a la planificació del desenvolupament del sistema de trànsit aeri europeu.

Nous modes de separació: separacions amb més informació disponible per ATCOs i eines ATC; o noves tècniques de separació basada en PTC avaluant el rendiment de navegació de l'aeronau, restriccions de l'espai aeri, i els temps d'arribades controlats per establir una precisa i altament predictibles seqüències d'enlairament i aterratge.

System Wide Information Management (SWIM): representa un canvi complet de paradigma de com es gestiona la informació, aportant beneficis directes tot el sistema ATM europeu, subministrant informació de qualitat. Es basarà en un marc per a l'intercanvi d'informació entre els agents tals com companyies aèries, operadors de rampa, ATC, operadors d'aeroports, etc. Incloent dades estandarditzades, i un conjunt d'enfocaments tècnics i no tècnics)

Automatització de processos: En el nou concepte de les operacions hi hauran moltes eines automàtiques per fer-se càrrec d'algunes de les tasques del responsable del tractament o ajudar-los amb la presa de decisions.

1.3.2. Identificació àrees de millora

Un cop s'han exposat les característiques de l'ATM actual i del futur model d'ATM (basat en el model que proposa SESAR) i se'n ha extret un anàlisi dels problemes observats i, es procedirà a determinar, d'una forma conceptual, les diferències principals entre els dos ambients de la gestió del tràfic aeri, avaluant els factors esmentats, identificant les àrees que es veuran afectades davant les intencions de millora proposades:

Operacions dels usuaris de l'espai aeri:

- en la planificació de les operacions de vol passa de basar-se en rutes d'aeronaus, a basar-se en **trajectòries**.
- A diferència de les rutes, les trajectòries **s'actualitzen constantment** durant el vol.
- La **navegació i comunicació** avançada esdevindrà, el requerit equipament del futur.

- **L'auto-separació dels avions** passarà la càrrega de treball del ATCO a la tripulació de la cabina.
- L'espai aeri estarà **dinàmicament estructurat** per objectius especials
- La introducció de **blocs funcionals** de l'espai aeri serà en prevenció de la fragmentació del espai aeri.

Operacions en xarxa:

- La gestió de trajectòries connectarà la **gestió estratègica** de flux amb les **operacions tàctiques** del ATC.
- Trajectòries 4D **altament previsibles** seran usades en contraposició de les rutes menys previsibles.
- Resolució de conflictes serà proporcionat **a nivell estratègic**, en lloc del nivell tàctic. Això disminuirà la càrrega de treball del controlador.

Operacions de l'aeroport:

- Els processos de CDM milloraran la **predicibilitat** i tindran un impacte positiu en el procés de gestió del flux.
- Ús més eficient dels recursos i d'una **xarxa més equilibrada**.
- Les pistes i carrers de rodatge seran usats en una manera més eficient causa dels nous procediments que permetin a menor **mínimes de separació durant l'aproximació**
- Superfície de moviment avançat de guia i control de sistemes **reduirà la càrrega de treball** del controlador i garantirà una **major capacitat de l'aeroport** en condicions de baixa visibilitat
- El entorn dels avions estarà gestionat amb **més precisió**, amb restriccions de temps que es connectaran directament amb trajectòries d'aeronaus.

Sistemes automatitzats:

- L'automatització es farà càrrec d'algunes de les tasques del controlador que **alliberarà recursos humans** per a les tasques, i serà d'ajuda per la presa de decisions, planificació i negociació.
- Seran necessaris nous sistemes automatitzats en la coberta de vol per a propòsits de separació d'aeronaus.
- Comunicacions entre terra-aire es convertirà en principal interruptor tecnològic per a nombrosos sistemes automatitzats.
-



Figura 8: Principals millores en les àrees de capacitat, seguretat, impacte ambiental i eficiència.

2. CONCEPTES PRINCIPALES DE LA PROPOSTA

En aquest capítol es pretén donar a conèixer una premissa d'informació relativa als temes directament relacionats amb el problema tractat: explicació del concepte de sistemes multi-agents i d'E-TCAS (dos dels mòduls inclosos en la plataforma). També s'obtindrà un coneixement sobre el projecte global (AGENT) que supervisa el disseny del Open Demonstrator.

2.1. Sistemes de *cockpit*: TCAS 7.1 vs. E-TCAS

TCAS II, va ser dissenyat per operar en tràfics de densitat major a 0.3 aeronaus per NM quadrada, millorant la seguretat en possibles conflictes però segueix tenint una **sèrie de mancances d'actuació en possibles escenaris de col·lisió**.

Quins són els punts de millora del E-TCAS respecte el TCAS 7.1?

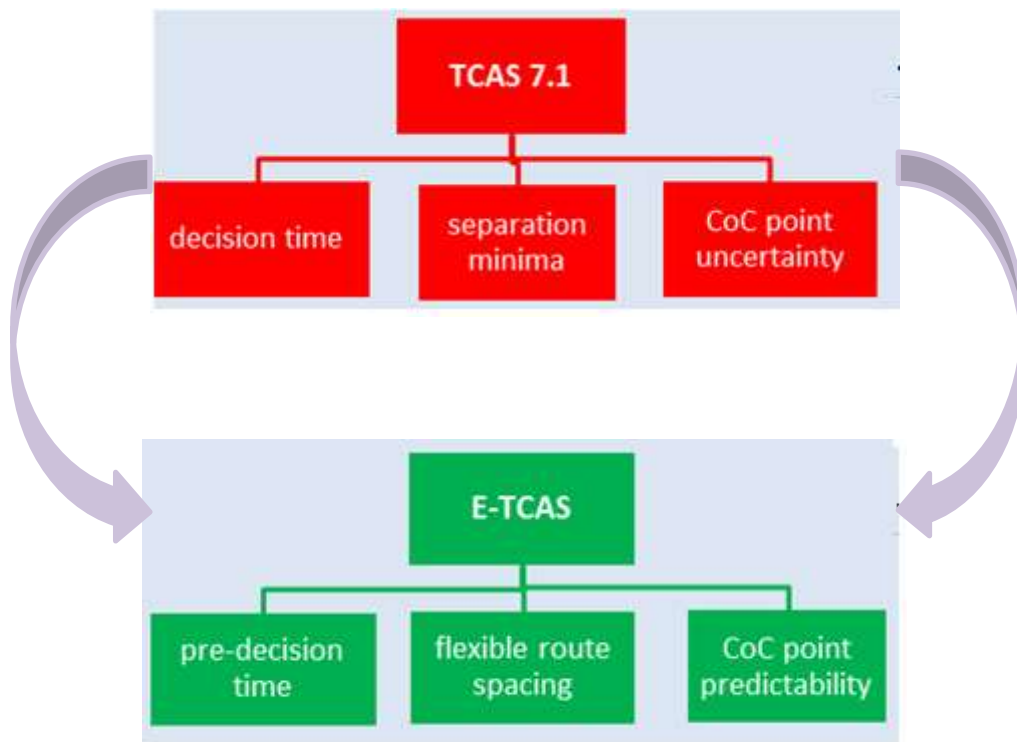


Figura 9: Representació esquemàtica de les diferències principals entre ambdós sistemes

Enhanced Traffic Alert and Collision Avoidance System (E-TCAS) pretén:

- Detectar i resoldre conflictes amb més nombre d'aeronaus
- Contemplar posicions de 4 dimensions.
- Un nivell superior de sensibilitat i alerta per TAs i RAs.
- Més seguretat de tram, més precisió en el punt de conflicte.
- Minimització del temps total de resolució.
- Facilitar les estratègies de maniobra
- Més atenció en el ascens i descens de nivells.

2.2. Comunicació dinàmica: *Multi-agent system*

Un **agent intel·ligent** és un programa/màquina que actua per aconseguir el millor resultat que es pugui per un objectiu concret, en base un criteri d'èxit del comportament. Tot i l'avanç de la tecnologia, hi ha **tasques complexes que un agent individual no pot resoldre de forma eficient**.

Un **sistema multi-agent** es refereix a un sistema format per múltiples agents intel·ligents semi autònoms que es relacionen i interaccionen entre ells en un mateix entorn, per **resoldre problemes treballant de forma conjunta**.

Característiques dels sistemes multi-agent:

- **Autonomia:** els agents són autònoms; no són dirigits per l'usuari o per altres agents.
- **Visió local:** els agents no tenen una visió global del sistema, sinó que només tenen accés a la informació d'una part de l'entorn.
- **Descentralització:** no hi ha un agent que controli el sistema.
- **Capacitat de comunicació:** els agents han de disposar d'un mecanisme comú per compartir informació (coneixement) entre ells.

- **Flexibilitat:** són molt flexibles donat que es poden modificar, afegir i reconstruir sense la necessitat de redefinir tots els seus components.
- **Auto-organització:** poden manifestar capacitat d'auto-organització i tendir a trobar la millor solució del problema amb poca o cap intervenció de l'usuari.
- **Comportaments complexes:** poden tenir comportaments complexos globals, encara que les estratègies individuals de tots els agents siguin simples.

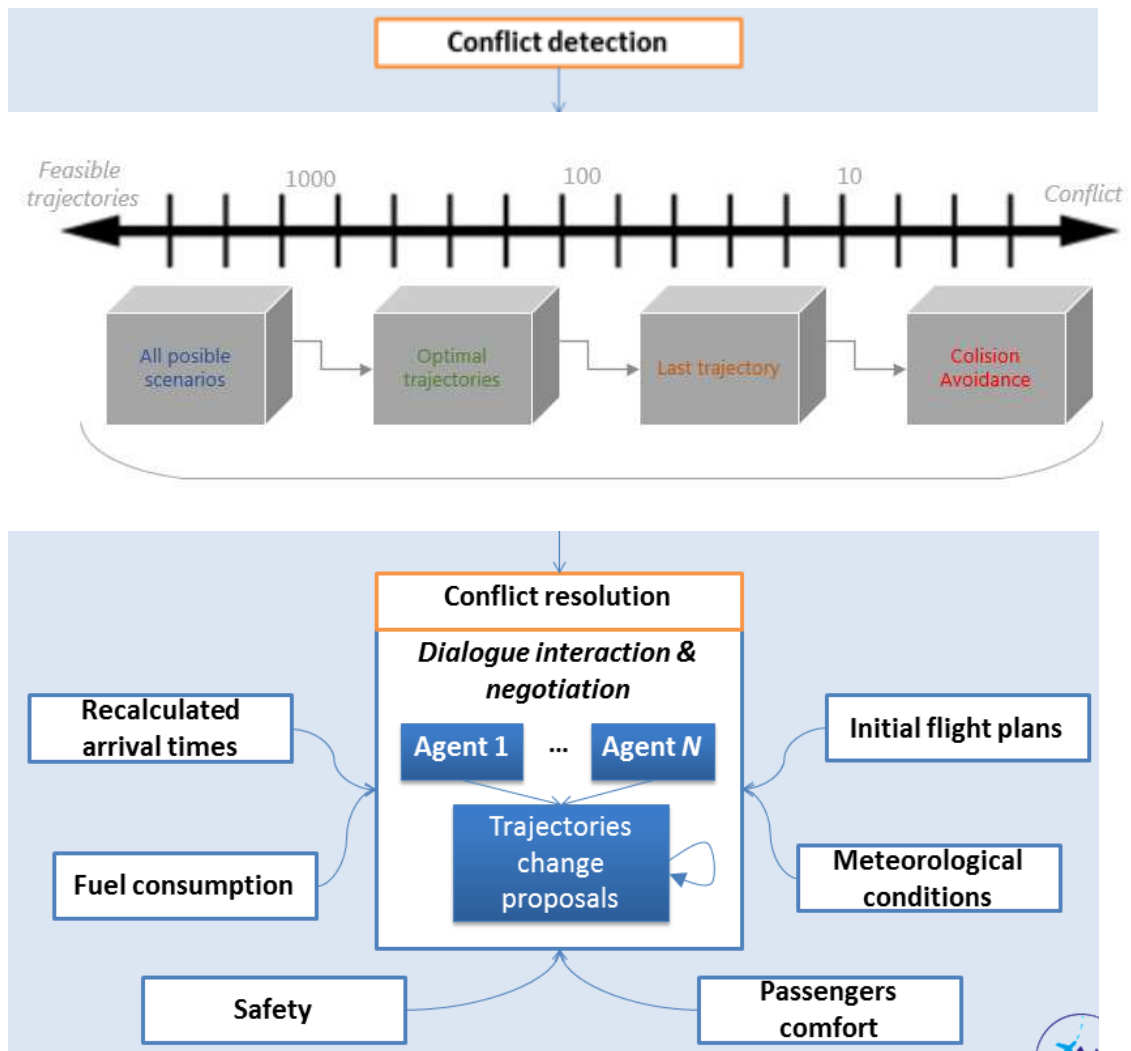


Figura 10: Model relacional en representació a les relacions entre els diferents agents i elements

La comunicació entre agents permetrà que cooperin, coordinin les seves accions i portin a terme tasques conjuntament. **Sense comunicació els agents són elements aïllats.**

La interacció entre agents es dona quan dos o més agents es relacionen a través d'un conjunt d'accions recíproques. Les conseqüències de les accions tenen una influència en el comportament futur dels agents.

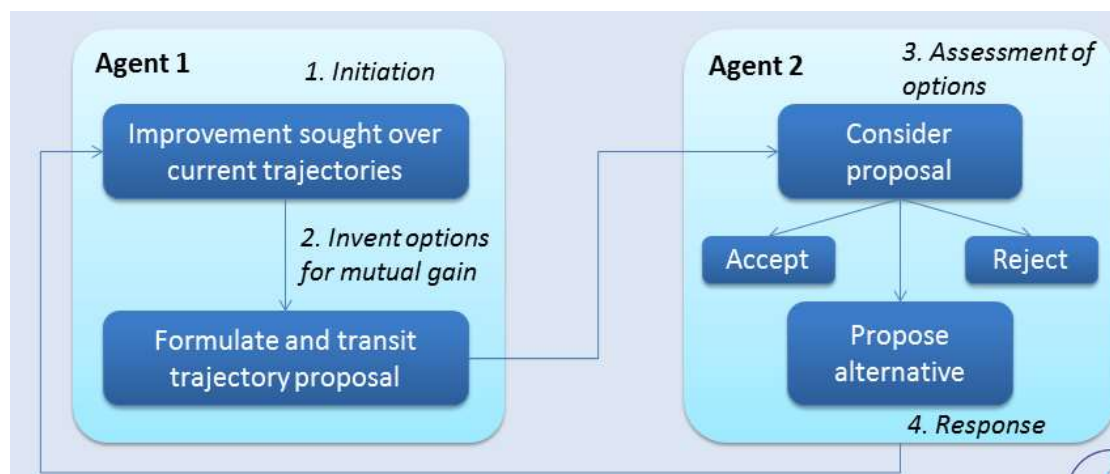


Figura 11: Exemple de procés de comunicació entre dos agents

2.3. Projecte AGENT: *(Adaptive self-Governed aerial Ecosystem by Negotiated Traffic)*

Després de l'explicació detallada dels conceptes teòrics bàsics, s'ha pogut obtenir un coneixement de les principals eines i sistemes més importants en l'operació de l'ATM, i també una àmplia percepció de l'evolució, situació actual i futura esperada de la gestió del tràfic aeri europeu. Davant d'aquest panorama il·lustrat, se'n pot treure un breu resum de les conclusions finals: **El ATM actual està buscant noves solucions i nous sistemes que permetin afrontar les noves expectatives de demanda.**

Actualment **SESAR** representa la empresa més efectiva ja que és el motor que permetrà integrar aquestes peticions a la gestió del espai aeri d'una

manera eficaç i ràpida, ja que **representa a la majoria d'institucions dedicades a la I+D del sector aeronàutic europeu**. Aquestes empreses o entitats tenen la característica comuna que han desenvolupat o estan participant en exitosos projectes, com per exemple: STREAM , LANTAK, INTERACTION, FUPOL, ANASTASIA, ALICIA, ACROSS, DARWIN, etc. entre d'altres.

La majoria d'aquests projectes estan enfocats a l'aportació de noves eines, tecnologies, algorismes d'optimització, sistemes o simulacions, que permetin resoldre les **àrees de millora** que el programa SESAR proposa.

Entrant en matèria, aquesta tesi es focalitza en la participació en la creació i desenvolupament d'un projecte (concretament, en una de les parts d'aquest) amb una intenció paral·lela als citats anteriorment. **Aquest projecte ha rebut el nom de AGENT**. Per entendre el desenvolupament dels temes tractats en apartats posteriors a aquest, cal entendre el context que els engloba.

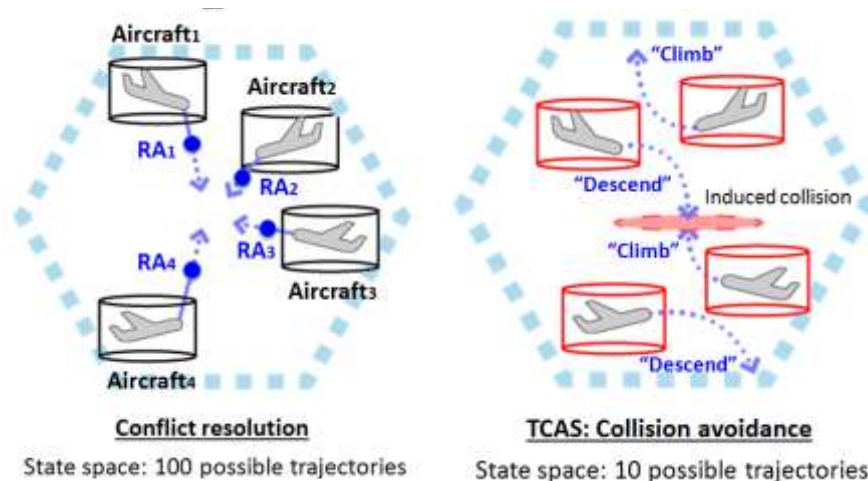
A continuació, es realitzarà una petita introducció dels aspectes més rellevants que envolten i que permeten assolir una idea inicial del projecte AGENT: **descripció general** per entendre la seva naturalesa i el seu paper en el panorama del ATM actual, els **objectius principals** que busca aconseguir amb la seva implementació i quin és **el punt en el que es troba** (és a dir, el punt que parteix aquesta tesi) i quin és **el punt que es vol arribar**.

2.3.0. Visió i propòsit de creació

El projecte AGENT es basa en el desenvolupament d'una plataforma basada en la **definició d'ecosistema dinàmic i autònom** que englobi diferents tràfics, i que proporcioni mecanismes i eines per a la **prevenció de possibles col·lisions** induïdes entre aquests. D'aquesta manera tan aviat com es detecti un conflicte proposi diferents resolucions basades en negociació i resolució de conflictes. Els diferents escenaris que es consideraran en la seva execució

són: conflictes entre aeronaus, condicions del temps, àrees restringides i condicions no planejades.

Per tant, el **projecte es basarà en E-TCAS**, per actuar a un nivell pre operacional, com un sistema d'anti-col·lisió robust, considerant escenaris on el **comportament humà i els aspectes d'automatització** tindran un paper important.



a) Els tràfics entren en zona de conflicte a un nivell RA

b) El TCAS actua davant d'una possible fase de col·lisió induïda.

Figura 12: Anàlisi de les trajectòries alternatives existents davant d'un escenari compost per 4 tràfics diferents.

En la figura de l'esquerra es pot observar com es crea un ecosistema integrat per quatre aeronaus amb E-TCAS en el moment que es preveu un conflicte, i es destrueix una vegada que les aeronaus surten de la zona de conflicte. Amb l'E-TCAS es proposa d'estendre fins a un **marge més gran de temps**, que permetrà l'exploració i anàlisi de més trajectòries possibles, en lloc de limitar-se a un nombre de trajectòries factibles petit i uns costos són alts. A la figura de la dreta es pot observar com la quantitat de trajectòries factibles identificades està disminuint a mesura que l'horitzó de temps avança. Cal comentar que davant d'una situació sense sortida, **també es disposarà la participació de l'ATC per resoldre els conflictes**.

2.3.1. Objectius AGENT aplicats al *Air Traffic Management System*

Per fer front als reptes identificats que actualment limiten l'automatització en el ATM, el projecte AGENT abordarà els següents objectius principals:

- Proposar un disseny innovador basat en l'automatització d'un futur sistema ATM per donar suport a un **canvi de sistema ATM avui dia centralitzat a un sistema distribuït**, en el qual les aeronaus estan formant els ecosistemes dinàmics.
- Desenvolupar eines de presa de decisions, utilitzades tan per ATC com pilots, per transformar els **avions en agents intel·ligents**.
- Construir l'ontologia per a la representació del coneixement, raonament i comunicació de màquina a màquina entre agents intel·ligents, permeten **l'intercanvi d'informació entre agents** que interactuen en un mateix ecosistema dinàmic.
- **Dinamitzar la operació del ATM** facilitant eines i tècniques relacionades amb *Big Data*, com la mineria de dades, visualització, processament de flux, etc.
- Verificar les eines de presa de decisions d'AGENT utilitzant la informació de trànsit real i per la **correcta validació dels resultats** per mitjà de diversos escenaris en un entorn de simulació.
- Demostrar i quantificar el potencial per al disseny automàtic innovador per proporcionar beneficis en matèria de **seguretat, la capacitat, l'eficiència i la seguretat de les operacions d'ATM**.



SECURITY



SAFETY



CAPACITY



COST-EFFICIENCY

Figura 13: logotips oficials respecte seguretat, capacitat i eficiència.

2.3.2. Resum abast de AGENT

Per fer front als seus ambiciosos objectius d'innovació, el projecte AGENT es basarà en els resultats d'anteriors projectes d'investigació, ja que seran clau per perfeccionar resultats clau que proporcionaran la base per al **desenvolupament i perfeccionament de les eines de CDM**.

Els resultats d'aquests projectes anteriors han demostrat la validesa de molts dels conceptes que es sustenta AGENT. El seu paper en aquest context serà en primer lloc **harmonitzar i madurar aquests resultats d'estudis**, a través d'un desenvolupament sistemàtic i un programa de proves. També haurà de fer **front a un augment de la demanda de trànsit i els nous reptes empresarials**. El projecte AGENT està alineat amb l'estratègia *UE 2020* que se centra en la construcció de l'economia europea en sòlids fonaments del coneixement, la recerca i la innovació.

El projecte construirà un *Open Demonstrator* disponible per a una àmplia comunitat científica i acadèmica com a laboratori d'assaig. Així, el projecte contribueix al **desenvolupament i avaluació d'idees no convencionals, conceptes, mètodes i tecnologies** que poden definir i oferir el rendiment necessari per a la propera generació del sistema europeu de ATM. D'aquesta manera, els resultats del projecte també contribuiran a **estimular la investigació científica** en solucions d'automatització autònomes, especialment tenint en compte un augment significatiu en el nombre i tipus de vehicles aeris en funcionament, inclosos els vehicles dirigits per control remot.



Figura 14: Emblema oficial del projecte AGENT

Per tant, aquestes són algunes de les moltes raons que afirmen que es tracta d'un projecte amb una ambició i aspiracions de gran envergadura.

3. ANÀLISI I DISSENY DEL *OPEN DEMONSTRATOR*

En aquest capítol es presenta la fase de definició, anàlisi i disseny de l'aplicació, part inicial i de gran importància de tot projecte basat en un producte de software, on es defineixen els requisits i es mostra una vista global de l'arquitectura pensada per al sistema.

Aquesta arquitectura s'analitzarà i s'exposarà des de diferents perspectives, que aniran evolucionant en funció de la renovació d'idees que es realitzin, és per aquest motiu, que possiblement la representació final hagi variat en referència a la inicial, degut a que aquesta s'ha anat transformant durant el mateix procés de disseny.

3.1. Anàlisi previ de les condicions

Posteriorment, es procedirà a la realització del disseny de tots els aspectes de l'aplicació, en base als requeriments especificats. És, per tant, en aquest punt on s'han definir els **fonaments del projecte**, dels quals es parteix per desenvolupar la resta. El procés a seguir es basa en primer lloc en triar una bona metodologia de desenvolupament d'acord amb el projecte.

3.1.1. Metodologia i procediments de l'elaboració

La metodologia de desenvolupament usada és una versió “semi-ajustada” al model iteratiu. Aquest tipus de model es basa en la repetició consecutiva de les següents tasques: anàlisi, disseny, implementació i proves, però aquesta tesi es centrarà només en les dues primeres fases: **anàlisi i disseny**. Això no significa que el desenvolupament del projecte general s'aturi en el moment que finalitzi aquesta tesi; **el projecte seguirà una evolució posterior**.

El procés de desenvolupament que es mostra en aquest capítol és el següent:

- Definició de l'aplicació i de les tecnologies que es necessitaran per l'elaboració del disseny i futura implementació.
- Explicació detallada de cadascun dels requisits funcionals de l'aplicació.
- Casos d'usos en representació dels requisits prèviament detallats.
- Primer prototipatge de la interfície gràfica d'alt nivell conjuntament amb el disseny conceptual de l'arquitectura de l'aplicació donant informació partint de diferents visions.

3.1.2. Finalitat de l'aplicació

El desenvolupament de la present aplicació de software es basarà en la creació d'una plataforma que es podrà utilitzar per provar futurs conceptes, com nous mètodes i tècniques, destinats a la **millora de la gestió del tràfic aeri**, validats en un entorn de simulació on es treballa amb dades reals, amb l'objectiu de facilitar una sèrie de processos i conseqüentment reduir la càrrega de treball del controlador aeri.

Aquesta aplicació ha de tenir una sèrie de prestacions i característiques per tal de complir el objectiu del projecte, és a dir que estigui **correctament orientat als usuaris** perquè realment pugui ser aplicat a l'àrea del problema.

Aquesta plataforma és actualment una **aplicació de laboratori**, però el resultat esperat es desplegarà en la **web**, permeten que els usuaris puguin accedir-hi a través d'un registre. D'aquesta manera s'augmenta la portabilitat i disponibilitat de la informació de l'aplicació, ja que prescindirà d'instal·lació.

Un cop acabat el procés, la usabilitat de l'aplicació serà provada en diferents moments, amb uns marges adequats per anar fent correccions sobre la marxa.

Encara que possiblement es tracti d'una aplicació de **codi tancat**, aquest és un tema que encara s'està investigant i s'acabarà de decidir durant l'evolució del projecte. En tot cas, el software final i tindrà el **suport d'una documentació** per facilitar processos i poder seguir adaptant-lo en funció de les futures necessitats.

3.1.3. Utilització de les tecnologies de desenvolupament

Al llarg del procés d'elaboració i representació del projecte s'han utilitzat una sèrie de **eines i tecnologies** que han facilitat la comprensió del desenvolupament realitzat. Referent a les **representacions gràfiques**, s'ha utilitzat:

Microsoft Visio: és un programari de dibuix vectorial per a Microsoft Windows. Microsoft va comprar la companyia Visio en l'any 2000. Les eines que el componen permeten realitzar diagrames d'oficines, diagrames de bases de dades, diagrames de flux de programes, UML, i més.. En aquest cas s'utilitzaran els diagrames UML, sobretot per diagrames de procés i diagrames de cas d'ús.

WireFrameSketcher: Aquesta aplicació proporciona moltes plantilles per dissenyar esbossos en molts sistemes diferents. En aquest cas, s'utilitzen les plantilles de web application. La seva interfície permet arrossegar els elements gràfics desitjats a la finestra de treball per formar la interfície necessària. Permet canviar el text contingut en aquests mateixos elements i afegir elements externs. Al treballar amb Eclipse, aquesta aplicació permet una bona integració amb l'IDE.

Referent a les tecnologies de **desenvolupament**, les eines més utilitzades per tal de processar i manipular les dades, han estat les següents:

- Per la configuració de la part de back-end (servidor):

Eclipse IDE: aquesta eina és un entorn integrat de desenvolupament de codi obert per programar principalment en Java, però també permet desenvolupar projectes en C, C++, COBOL, Python, Perl, PHP i altres. En definitiva, Eclipse és una interfície gràfica completa i fàcilment adaptable a les necessitats de l'usuari, que permet treballar d'una forma còmode i flexible.

llenguatge de programació Java: El motiu d'elecció d'aquest llenguatge és bàsicament per la seva senzillesa i flexibilitat. Aquest llenguatge orientat a objectes, es caracteritza per la seva concurrència alhora d'executar diferents fils d'execució i per la seva fàcil distribució del codi en classes i objectes, propietat necessària per tal d'incorporar fàcilment noves funcions al codi. A més a més, és un llenguatge d'ús molt generalitzat, per tant es poden buscar diferents recursos, llibreries o funcions concretes a través de la xarxa, que resolguin dubtes i solucionin les necessitats que vulguem satisfer durant el desenvolupament de codi.

Tot i així, la futura plataforma es caracteritzarà per la possibilitat d'integració i adaptació de diferents mòduls, desenvolupats en diferents llenguatges de programació (C/C++, Python,.NET, etc.), per aquesta raó possiblement esdevingui una **aplicació multi-format**.

- Per la configuració de la part de front-end (client):

HTML (Hyper Text Markup Language): Degut a l'èxit del nou HTML5, seria una bona opció el seu ús. és un llenguatge d'etiquetes dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents per a la web (front-end web).

JavaScript: La seva elecció és degut a la seva compatibilitat amb altres llenguatges, i per ser un llenguatge força estructurat i funcional. És un llenguatge script amb gran varietat d'aplicació. És conegut sobretot pel seu ús en pàgines web, però també s'utilitza en altres aplicacions, amb un objectiu d'ús de millora de interfície. Les noves llibreries de JS, com AngularJS o Backbone.js, ens poden oferir solucions ràpides amb molts possibles problemes.

3.1.4. Definició dels requisits funcionals principals

Una de les fases del projecte ha estat l'elaboració d'una especificació tècnica tancada, on es descriuen els requeriments que ha de satisfer el producte. Els requisits funcionals s'acostumen a utilitzar per determinar quines **accions farà el sistema i quines restriccions pot tenir**.

A continuació es detallen alguns d'aquests requisits funcionals bàsics que determinen el **comportament de la plataforma** i que permetran obtenir els resultats que s'esperen de l'aplicació. (el document d'especificacions i requeriments complet es trobarà adjunt a la secció d'annexos de la memòria). Existeixen altres requisits iguals o més importants per tal de descriure l'aplicació, anomenats **requisits no funcionals**, però en aquest apartat és donarà èmfasi només els que representen exclusivament el funcionament.

Els diferents requisits funcionals de la plataforma es poden dividir en dues parts: la gestió d'usuaris i l'activitat pròpia de l'aplicació:

Gestió d'usuaris

RF1: Registre

- L'aplicació ha de permetre al usuari introduir les seves dades en un formulari de registre.
- El sistema s'encarregarà de validar les dades.

- El sistema mostrarà un missatge d'error si alguna informació introduïda no és correcta amb les condicions establertes en el formulari.
- En cas que sigui correcta, el sistema guardarà les dades del usuari en una base de dades.
- En cas que sigui correcta, el sistema enviarà un correu de confirmació per justificar el correcte registre.

RF2: Inicialització de la sessió

- Per iniciar la sessió el usuari haurà d'identificar-se amb el nom d'usuari i contrasenya corresponent. En el cas d'haver iniciat sessió anteriorment, el inici de sessió serà automàtic al iniciar l'aplicació.
- El sistema s'encarregarà de validar i permetre o denegar el accés a l'aplicació.
- El sistema mostrarà un missatge d'error en el cas que la validació no sigui correcta.
- Si no hi ha hagut problemes en el procés d'accés, es redirigirà a la pantalla principal de l'aplicació.

RF3: Finalització de la sessió

- Qualsevol usuari de l'aplicació ha de poder finalitzar la sessió en l'aplicació amb un botó on s'indiqui "tancar sessió".
- En el cas que usuari premi el botó de tancar sessió, el sistema mostrarà un missatge per assegurar-se que l'usuari vol tancar la sessió, donant la opció d'acceptar o cancel·lar.
- El usuari serà redirigit a la pantalla principal però en aquest cas sense accés a l'aplicació.

RF4: Compte d'usuari

- Ha d'existir una pantalla en el sistema que permeti al usuari consultar, modificar o eliminar les dades del seu compte.
- Si modifica alguna dada el sistema ho ha de validar.
- Ha d'existir una opció de donar-se de baixa dintre la configuració del compte.
- En el cas que la validació no sigui correcta, permetrà reescriure la nova informació, Indicant-li l'error.
- En cas que la validació sigui correcta, s'actualitzaran les dades a la base de dades.

Aplicació:

RF7: Pantalla inicial

- La pantalla inicial s'hi mostrarà una finestra d'accés central. Aquesta finestra et permetrà dues opcions:
 1. la opció de donar-se d'alta.
 2. En cas de registre previ, introduir nom d'usuari i contrasenya d'accés.

RF8: Zona central (pantalla inicial)

- Hi constarà un visualitzador que permetrà conèixer les eines que suporta la plataforma.
- Cada eina estarà representada amb el logotip de l'eina i una petita descripció del seu ús.
- El visualitzador estarà configurat en un format de catàleg, que permetrà canviar d'eina amb unes fletxes horitzontals.
- Cada eina estarà enllaçada amb una pàgina extensiva-

RF8: Menú lateral esquerre (pantalla inicial)

- En el menú lateral esquerre es descriuran les diferents accions que l'usuari pot realitzar:
 1. recerca i visualització de les fonts d'informació disponibles.
 2. incorporació de noves fonts d'informació.
 3. enllaç directe amb la pantalla d'elecció modular
 4. Coneixement d'informació extra relacionada amb l'aplicació

RF9: Menú lateral dret (pantalla inicial)

- Es mostrarà la opció de dades referents en el següent ordre:
 1. Gestor de missatges: enviats, rebuts i guardats.
 2. Els meus arxius: fitxers i informació necessària per a treballar.
 3. notificaciones d'actualització de les eines.
 4. Manuals d'ajuda i documentació referent a les eines.
 5. FAQ: Preguntes freqüents.
 6. Afiliació: Pantalla d'informació sobre plans d 'afiliació.
 7. Informació rellevant sobre l'empresa.

RF10: Barra de configuració superior (pantalla inicial)

- Es mostrarà el logotip de l'aplicació.
- Es mostrarà l'opció d'accedir a les dades del perfil del d'usuari.
- Es mostrarà la opció de tancament de la sessió.

RF11: Finestra visual inferior (pantalla inicial)

- S'hi incorporarà informació de notícies recents del sector, nous projectes, etc.

RF12: Pantalla general mòduls

- Es mostrarà un catàleg de tots els mòduls de funcionalitats que disposa l'aplicació.

- Donarà la opció de complementar resultats prèviament obtinguts amb altres eines, realitzar combinacions i proves.
- Incorporació d'una pestanya "modify modules" que permeti modificar alguns dels mòduls existents en la plataforma.
- Incorporació d'una pestanya "add modules" de càrrega de nous mòduls.
- Incorporació d'una pestanya "search modules", que permeti una recerca d'e funcionalitats per mitjà de la inserció de conceptes.

RF13: Finestra individual mòduls

- Permetrà la utilització, afegiment i inserció de noves eines en el mòdul seleccionat en funció dels requeriments i característiques d'entrada i de sortida de informació.

3.1.5. Representació diagrama de casos d'ús

Els **diagrames de casos d'ús** són una descripció dels passos per construir un procés. Per tan, per representar aquestes activitats diferents que l'aplicació realitza, s'utilitzaran els diagrames de casos d'ús. Aquestes eines et permeten identificar i esquematitzar les relacions entre els agents i els passos necessaris que calen per portar a terme alguna operació de l'aplicació.

Els **actors principals**, és a dir, entitats que mantenen una relació amb l'aplicació i que hi intervenen aportant una funcionalitat, són:

- Els usuaris beneficiaris del servei, que poden ser: SESAR, EUROCONTROL, *Airspace Users*, *ATC System*, *ATM community*, etc.
- Administradors del sistema: equip de desenvolupament, enginyers, usuaris participants, etc.
- Sistemes d'aprovisionament de dades: fonts d'informació.

Els **casos d'ús** identificats en els diferents requisits funcionals són:

- Utilització de les eines existents
- Modificació dels mòduls existents
- incorporació de nous mòduls
- incorporació de noves fonts d'informació

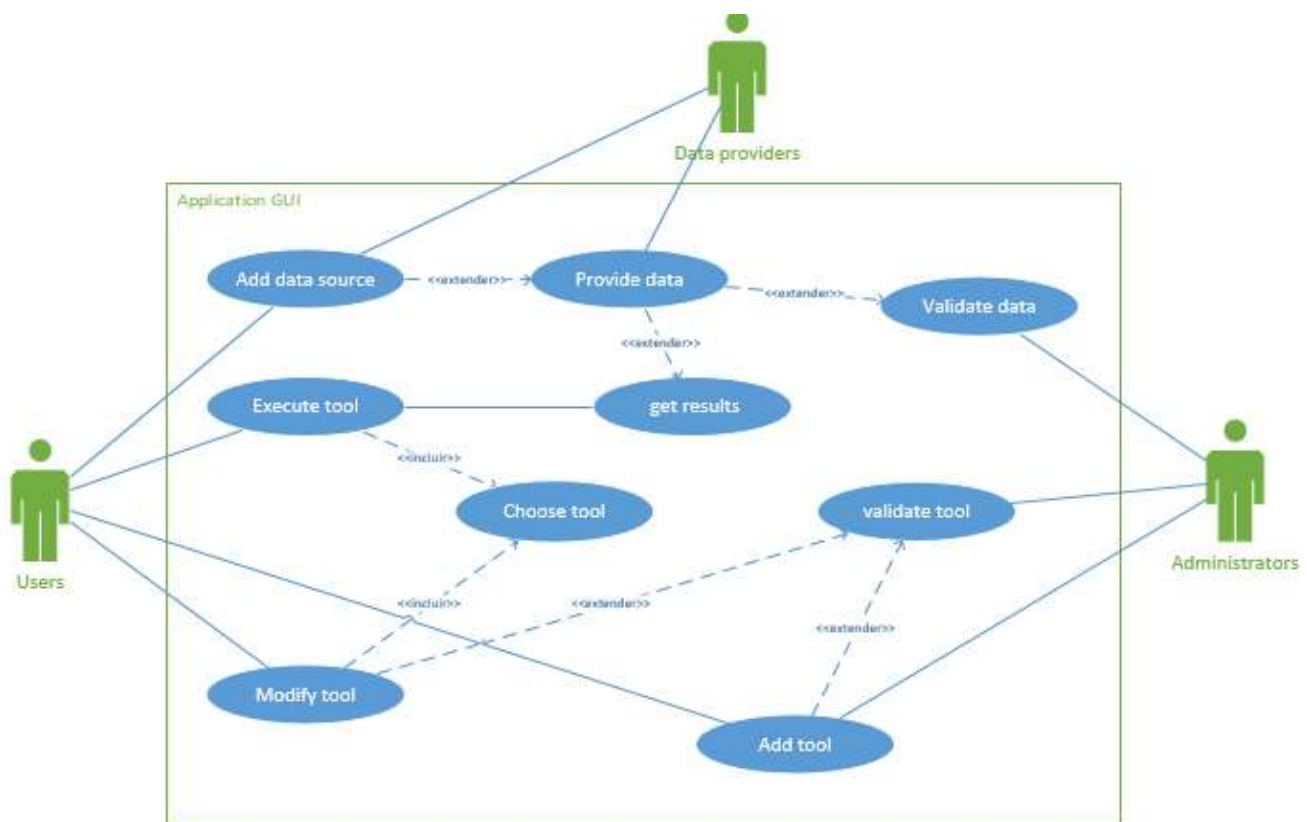


Figura 15: Model relacional en representació a les interrelacions entre els actors a través dels casos d'ús de l'aplicació.

2.2. Disseny conceptual i evolució del *Open Demonstrator*

Qualsevol projecte que es basi en la creació d'un programa o disseny de software, ha de seguir un anàlisi i disseny que proporcioni les bases en les que es desenvoluparà l'aplicació. Concretament començant amb l'elaboració d'un disseny de l'arquitectura general.

El **disseny arquitectònic** es basa en la descomposició i divisió en diferents subsistemes que, de forma conjunta, permet oferir un servei concret. Una arquitectura es caracteritza per la combinació d'estructura de programes i l'estructura de dades y defineix les interfícies que faciliten el flux de dades. El seu principal objectiu és que permeti obtenir una **visió general del software a desenvolupar**.

En aquest procés de disseny cal identificar aquests subgrups i establir un marc que permeti la comunicació entre ells, i la seva execució per parts.

Alguns dels **principals avantatges** que ens facilita aquest disseny són:

- Les decisions que es prenen en aquesta fase del projecte permet afectar directament a la **fiabilitat, manteniment, rendiment i operativitat** general del producte.
- Estructura fàcilment **modificable i adaptable** a altres components de software.
- Permet descriure clarament les estratègies.
- Utilització de diagrames que mostren la relació entre el **hardware** i el **software**.
- Permet mostrar visualment el impacte dels requisits funcionals, no funcionals i restriccions del sistema.

Per l'altra banda, **prescindir** del disseny arquitectònic del projecte podria comportar:

- Risc a construir un **sistema inestable** i no escalable.
- Un sistema que falli quan es realitzin petits canvis de correcció o millora.
- Un sistema que sigui **difícil de provar** i n'allargui el període de proves
- Un sistema que dificulti la seva gestió.
- La qualitat del sistema no pugui ser avaluada fins més endavant, quan quedi poc temps i ja sigui s'hagi realitzat la inversió indicada.
- **Sistema ineficient** que no s'optimitzin els recursos i que possiblement no s'ajustin a les necessitats del client.

Tota arquitectura de software ha de descriure diversos aspectes del software. És necessari conèixer les **diferents vistes** que es poden utilitzar per fer que aquesta descripció sigui més comprensible.

La visió estàtica: descriu quins components té l'arquitectura.

La visió funcional: descriu què fa cada component.

La visió dinàmica: descriu com es comporten els components al llarg del temps i com interactuen entre si.

El model concret d'arquitectura utilitzat en el projecte tractat, es tracta d'una combinació d'estructura per capes amb una arquitectura basada en una descomposició modular interna.

2.2.1. Visió estàtica dels elements arquitectònics

Abans de començar a desenvolupar el cos del projecte, s'ha de definir amb claredat l'estructura de disseny del *Open Demonstrator*.

Aquesta plataforma suportarà la unió de diferents conceptes relacionats amb un objectiu comú: facilitar una informació que permeti una previsió de conflictes entre trajectòries i conseqüentment, retorni una anticipació basada en la resolució dels conflictes.

A continuació es delimita una visualització gràfica prèvia en la que s'hi representa la constituent arquitectura modular del *Open Demonstrator*.

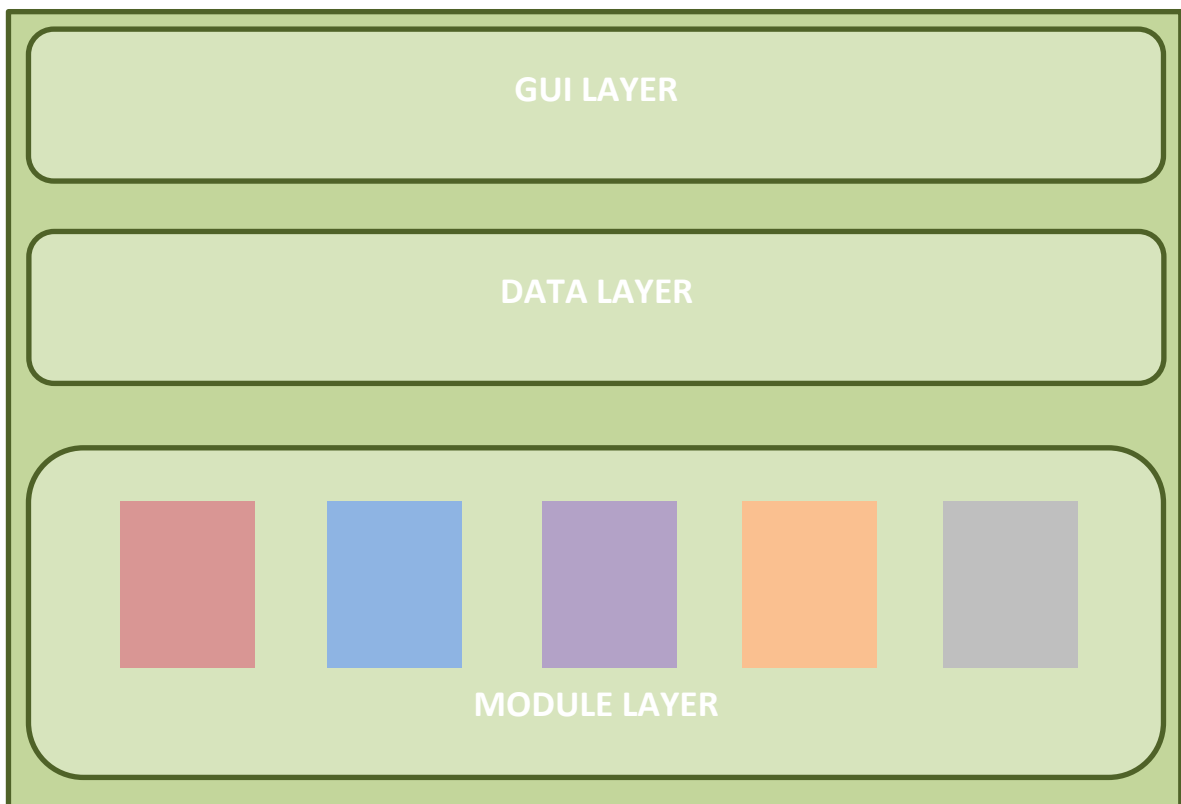


Figura 16: Representació esquemàtica de la visió estàtica de l'arquitectura

Com es pot observar, el model es caracteritza per la diferenciació de diferents nivells de capes. Cadascuna d'aquestes capes té unes característiques

particulars que permeten la incorporació de diferents elements, creades per realitzar una funcionalitat concreta dintre del funcionament conjunt.

2.2.2. Visió funcional dels elements arquitectònics

Tot seguit s'explicarà d'una forma més detallada les diferents propietats i especificacions internes de cadascuna de les capes.

Graphical User Interface Layer (GUI Layer): Una *Graphical User Interface* (GUI) és una interfície d'usuari que per mitjà de la utilització d'elements gràfics (sonors, visuals, de control, d'execució, etc.) permet interactuar amb un sistema informàtic d'una forma més fàcil i intuïtiva que no pas a través de les línies d'ordres, més difícils d'entendre i d'utilitzar. En el cas estudiat, la *GUI layer* es presenta el sistema al usuari comunicant-li la informació i capturant les respostes que l'usuari contesta. ha de ser fàcil d'aprendre i dominar. Aquesta capa només es comunica amb la *data layer*. Es tracta d'una capa que realitza la funció de incorporació de diferents interfícies d'usuari que permetin una fàcil interacció amb el contingut dipositat en la capa de mòduls a través de la capa de transmissió de dades. La seva principal característica és la capacitat d'incorporar noves eines buscant que cadascuna d'elles aporti un nou complement a la funció general de la plataforma. Existeixen exemples d'eines que són possiblement incorporables en aquesta capa, ja que se'n coneixen les utilitats i els possibles beneficis que podrien aportar amb la seva implementació. Aquestes eines són les següents:

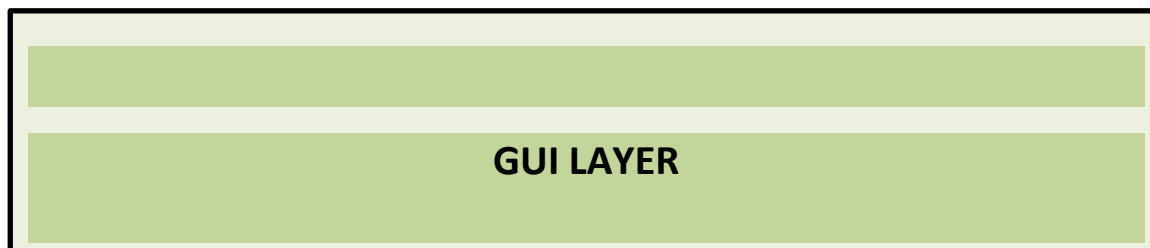


Figura 17: Il·lustració d'algunes components als qual es tindrà accés a través de la GUI layer

GUI Stratway. Interfície creada i relacionada directament amb l'eina *Stratway*, per tant, interfície bàsica en el desenvolupament d'aquest projecte. La seva funció principal és la de permetre al usuari una visió simulada del comportament d'un conjunt de trajectòries, i que per mitjà de diferents elements gràfics, com ara botons de càrrega d'arxius (.csv) sobre possibles condicionants que puguin afectar a l'evolució del vol, o configuració dels paràmetres que componen aquella dimensió del espai aeri. La seva funció s'acabarà d'especificar amb la descripció més concreta del mòdul de detecció i resolució de conflictes.



Figura 18: Representació gràfica del output de la GUI Stratway

CPN Tool. és una eina per a l'edició, simulació i anàlisi de xarxes de Petri d'alt nivell. És compatible amb les xarxes de Petri bàsiques, xarxes de Petri temporitzades i xarxes de Petri acolorides. Disposa d'un simulador i s'inclou una eina d'anàlisi dels possibles escenaris (*State Space Analysis*). L'eina disposa de comprovació de sintaxi incrementals i la generació de codi, que tenen lloc mentre s'està construint una xarxa. Un simulador ràpid gestiona de manera eficaç les xarxes sense hora i

cronometrats. Poden ser generats i analitzats espais complets i parcials d'estat, i es facilita un informe d'espai d'estat estàndard on conté informació, com ara les propietats d'acotació.

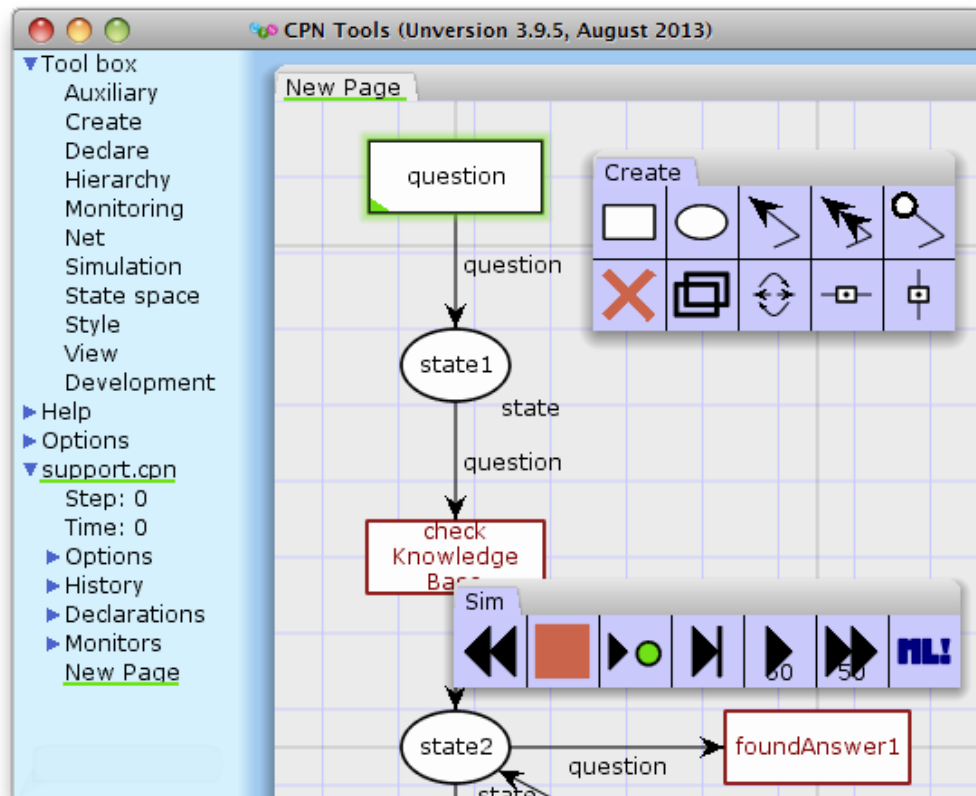


Figura 19: Captura de pantalla de la interfície de CPN Tools

L'aplicació pràctica d'aquesta eina és la de considerar un nou enfocament de model causal per a la divisió d'esdeveniments discrets davant del conflicte o col·lisió, tenint en compte una previsió temporal, que es descompon en una seqüència de temps de curts intervals per a les accions de control. El model es basa identificar un temps anticipat que ofereixi una predicció d'esdeveniments de conflictes i la manera en com evitar el punt calent. També busca un procés de pre-decisió que doni avantatge sobre un procés de presa

de decisions, pel que fa a les actuacions de l'aeronau. Amb una disminució contínua de les possibles trajectòries assolibles i lliures de conflictes.

InCAS (*Interactive Collision Avoidance Simulator*): és un sistema interactiu per a l'avaluació, els estudis, demostració i capacitació en suspensió en l'aire sistema anticol·lisió (ACAS). Fa la funció de un simulador flexible i totalment integrat a ACAS, incloent totes les característiques necessàries per a preparar, executar i analitzar simulacions ACAS de conflictes entre aeronaus, a partir de les dades obtingudes de radar. Permet preparar sèries de formats estàndards de dades de radar, configurar escenaris, executar o reproduir simulacions ACAS, veure un conflicte a través de pantalles pilot i el controlador simulats, analitzar i diagnosticar el possibles comportaments dels escenaris, i còpies en paper d'impressió de les diverses vistes d'una simulació d'ACAS.

NetLogo: És un entorn de modelatge programable multi-agent. Aquesta eina permet realitzar simulacions obertes, explorar com es comporten en diferents situacions i crear models personalitzats. És especialment adequat per modelar sistemes complexos en desenvolupament en el temps. Els modeladors poden donar instruccions a centenars o milers de "agents" que actuen tots de manera independent. Això fa que sigui possible explorar la connexió entre el comportament a nivell micro dels individus i els patrons de nivell macro que sorgeixen de la seva interacció. Es tracta d'un entorn de modelatge programable per a la simulació de fenòmens naturals i socials. *NetLogo* és prou simple i prou avançada per servir com una poderosa eina per als investigadors en molts camps.

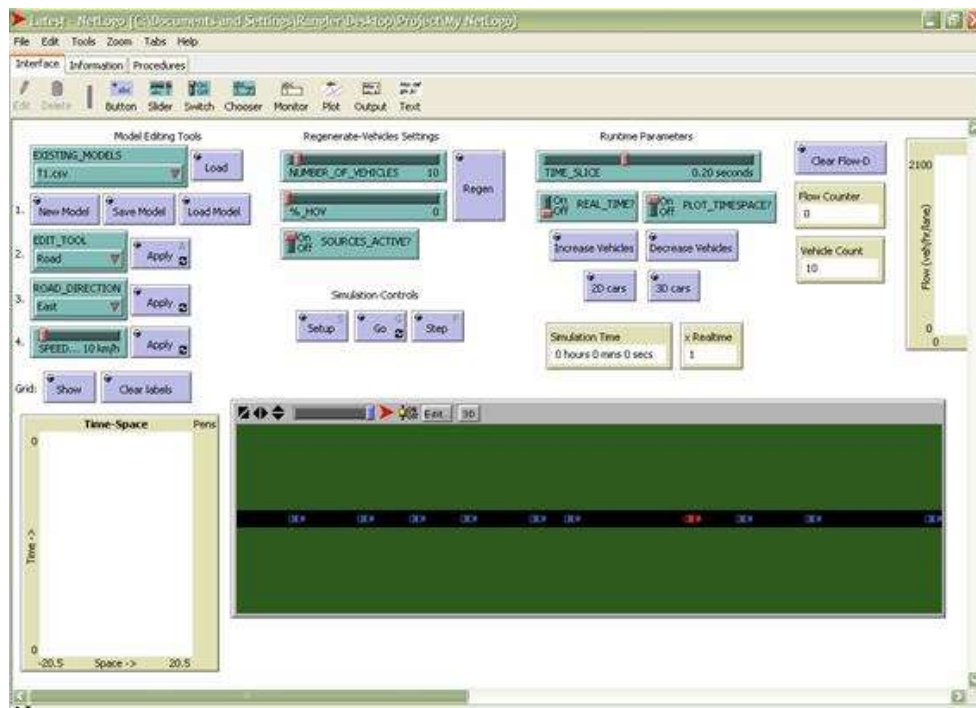


Figura 20: Captura de pantalla de la interfície de NetLogo

Existeixen altres eines igual de útils i incorporables que les descrites anteriorment, com ara *BlueSky*, *NEST*, *Understand* o entre d'altres, però la seva funció és prescindible i no estan oficialment assignades a la validació de la informació processada.

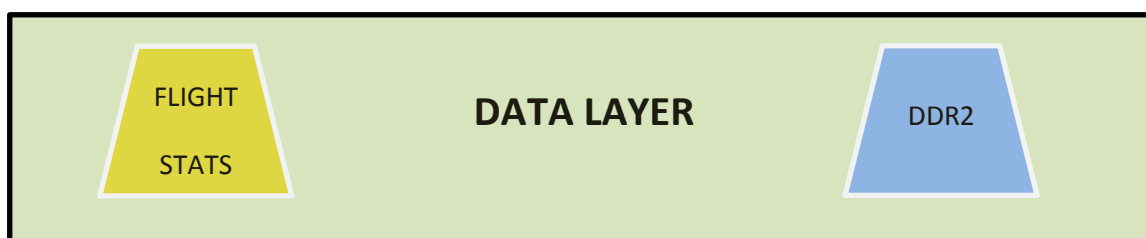


Figura 21: Representació dels components que inclou la Data layer

Data Layer: s'encarrega de l'execució dels programes interns, es reben les peticions del usuari i s'envien les respostes després de processar-les. Aquesta capa es comunica amb la capa GUI, per rebre les sol·licituds i presentar els

resultats, i amb la capa modular, per emmagatzemar o recuperar dades d'ella. es tracta d'una capa del programa que proporciona accés a les dades emmagatzemades. La seva principal funció en l'estructura del *Open Demonstrator* és la de transmissió d'informació. Representa les possibles fluctuacions de dades en les diferents direccions: informació processada per la capa modular que es transmet a la capa d'interfície gràfica per tal de mostrar-la a l'usuari, o en el sentit contrari, peticions realitzades per l'usuari per accedir a les dades de la capa modular.

Les dades que es fan servir per tals proves, són dades de plans de vols reals i sobre les condicions climàtiques. En aquesta tesi no es treballa amb dades referents a la meteorologia, però en la versió final si que es contemplarà. Les fonts escollides com a proveïdors d'aquesta informació són les següents:

FlightStats: és un cercador de vols, on s'hi inclouen les respectives planificacions i retards. Ofereix una gran varietat de informació relacionada amb el transport aeri, i et permet filtrar la informació en funció dels teus interessos i necessitats.



Figura 22: Captura de la pantalla d'inici de la font de dades FlightStats

Demand Data Repository 2 (DDR2): L'objectiu de DDR2 és proporcionar als usuaris de l'espai aeri dades de trànsit, trajectòria i medi ambient precises, per tal d'analitzar l'entorn de xarxa i la informació de capacitat. També es poden realitzar estimacions del futur trànsit. Ofereix diferents formats d'arxiu (SO6_m1,SO6_3,ALL_FT,EXP), en funció de la informació necessitada.

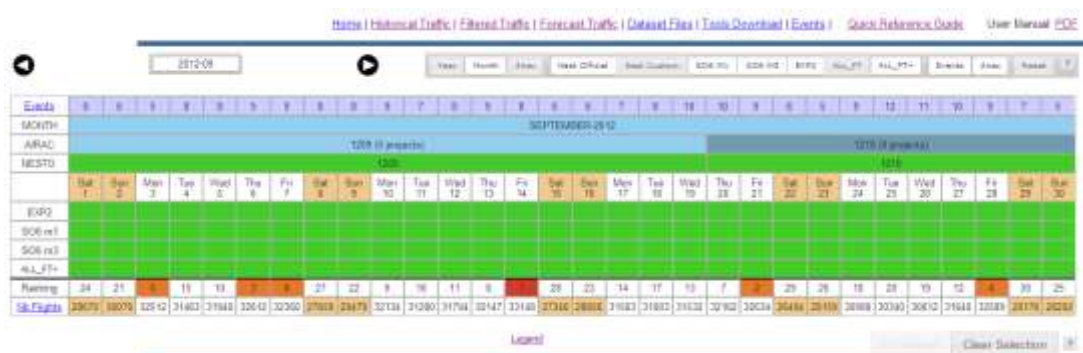


Figura 23: Captura de la gràfica d'arxius descargables a través de DDR2

Totes dues fonts tenen unes condicions similars: és necessari donar-se d'alta com usuari, el registre és gratuït (però limitat a uns serveis i a un temps), oferint un nombre concret de recerques de informació. La diferència principal és que en DDR2 es necessita que EUROCONTROL et doni accés a aquesta base de dades. En canvi, *FlightStats* és d'ús més generalitzat. Una vegada se'n ha registrat i es té accés a la informació, només cal crear les funcions necessàries per cridar aquests serveis de recerca des de la mateixa aplicació.

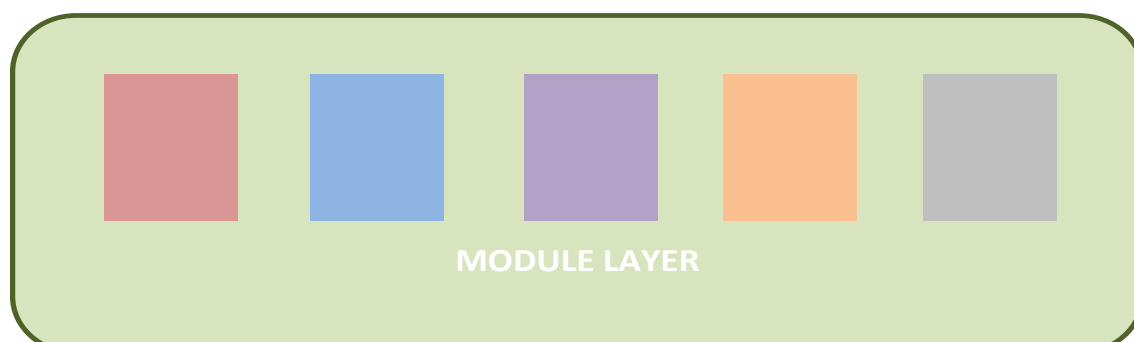


Figura 24: Representació dels components que inclou la module layer

Module Layer. És on resideixen les dades i es la encarregada de accedir a elles. Es realitza tot el emmagatzematge de dades, reben sol·licituds d'emmagatzematge, modificació o recuperació d'informació des de la *data layer*. Aquesta capa té la funció de permetre la adaptabilitat de diferents mòduls i de poder transmetre informació entre ells. Alguns dels mòduls que es podrien incorporar són els següents:

Conflict Detection & Conflict Resolution (CD&CR): Mòdul bàsic d'on s'extreu la informació proporcionada per les companyies aèries i altres fonts reals (trajectòries, condicions del temps...). Aquest mòdul té la propietat d'adaptació a diferents formats de dades. L'eina en la que es basarà el cas d'estudi: *Stratway*, forma part d'aquest mòdul. En aquest cas, la funció principal és la 'd'aplicar diferents algorismes que permetin detectar i resoldre d'una forma ràpida i automàtica les trajectòries en risc de col·lisió.

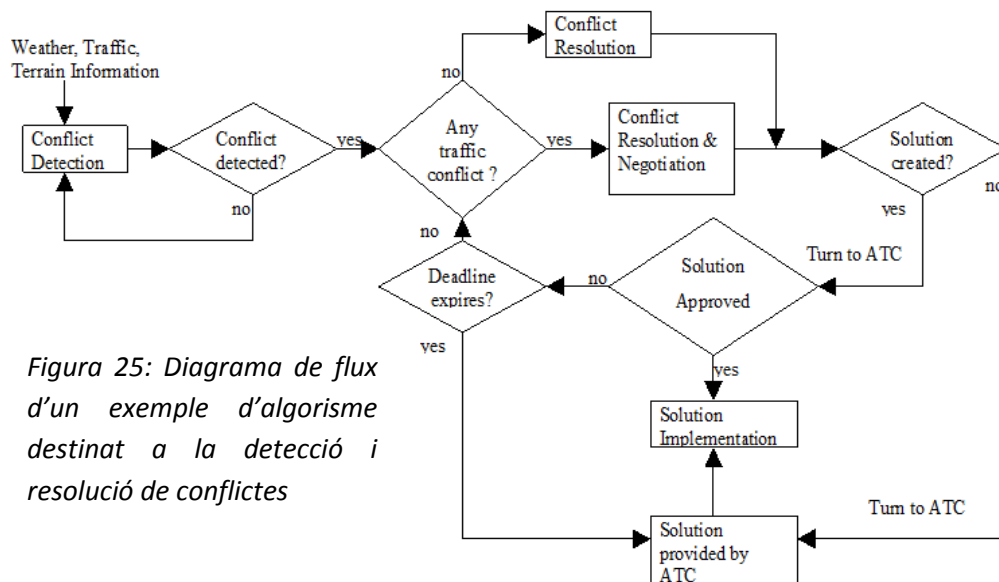


Figura 25: Diagrama de flux d'un exemple d'algorisme destinat a la detecció i resolució de conflictes

State Space Analysis. L'eina principal basada en aquest bloc es *CPN tools*. La idea bàsica del *State Space Analysis* és la de calcular tots els estats assolibles i els canvis d'estat del model CPN i representar a

aquests en una dirigida gràfica on els nodes es corresponen amb el conjunt d'estats assolibles i els arcs corresponen als esdeveniments. Per tant, un model en CPN pot ser desenvolupat com un enfocament clau per analitzar l'espai d'estats d'un escenari de trànsit congestionat en el qual s'exploren els esdeveniments que podrien impulsar una trobada en una col·lisió.

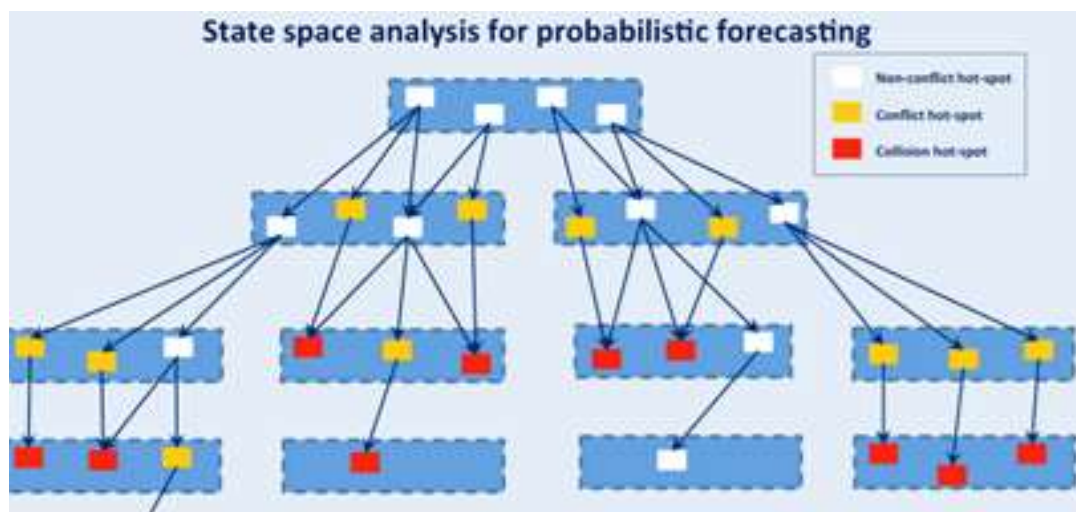


Figura 26: Graf d'exemple en l'anàlisi de diferents escenaris

Trajectory Negotiation: Aquest mòdul suporta el concepte d'una operació basada en un sistema multi-agent. Com ja s'ha dit anteriorment, aquest concepte significa l'existència d'agents que generen simulacions i prenen decisions d'una forma autònoma, adaptables a diferents entorns (ecosistemes dinàmics), i a la vegada, cooperatius. En aquest mòdul es permet la incorporació de informació referent a la interacció i la comunicació entre els diferents agents, sense requerir la intervenció ATM.

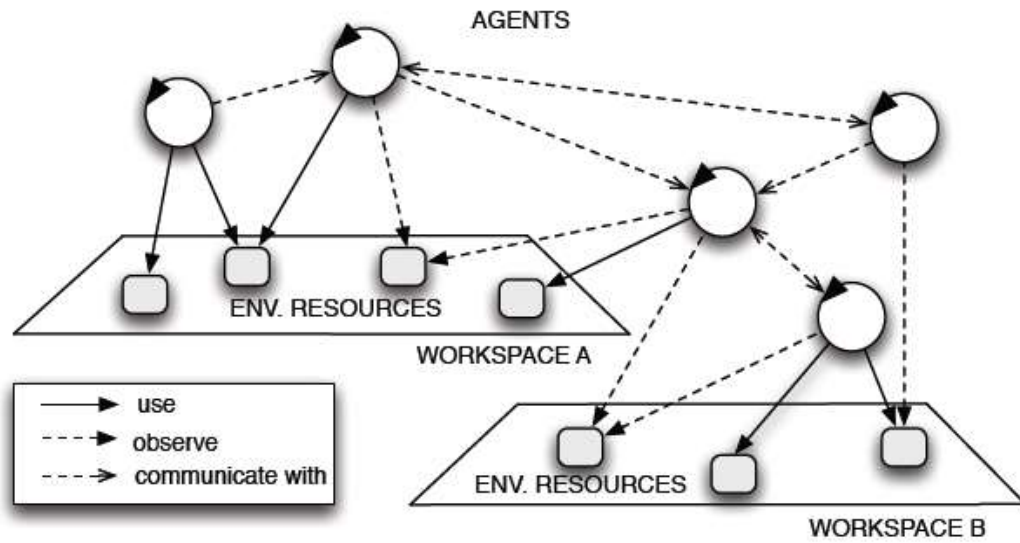


Figura 27: Esquema il·lustratiu d'una comunicació i connexió entre diferents elements

col·lisions entre diferents aeronaus i funciona independentment als serveis de trànsit aeri. La informació que retorna té la funció d'evitar una col·lisió en la fase que comprèn els últims segons abans de la possible col·lisió. En aquest mòdul es recull informació d'aquest sistema d'anti-col·lisió de tràfic aeri, sobre els vols vigents per tal de reduir els incidents que es puguin realitzar a mig vol entre aeronaus.

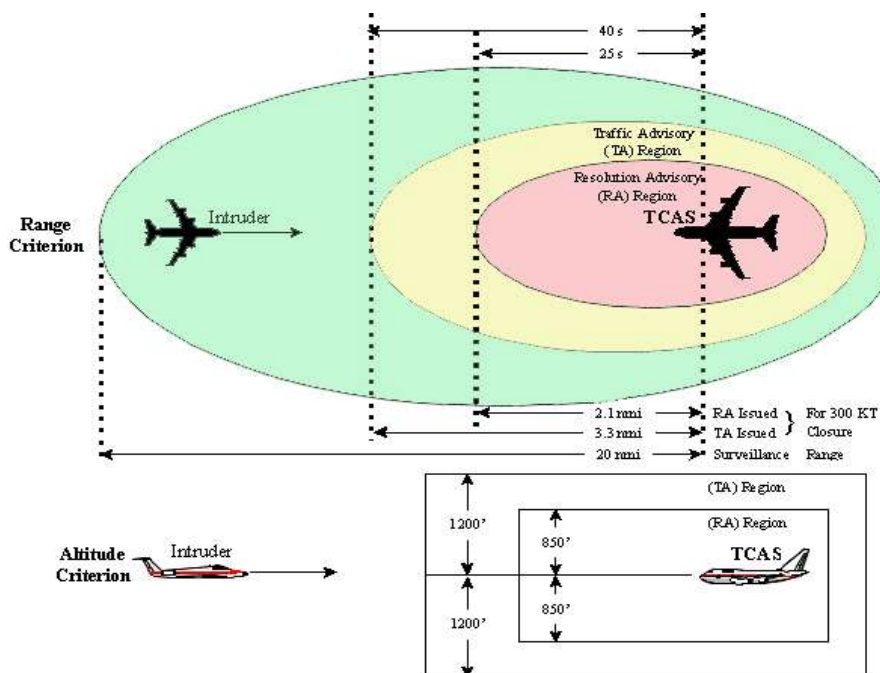


Figura 28: Il·lustració del funcionament del sistema TCAS

KPI (Key Performance Indicators): són indicadors que mesuren el nivell de rendiment d'un procés concret. Aquest valor (normalment en percentatge) està relacionat amb algun objectiu fixat. La incorporació d'aquest tipus de mòdul permet la introducció d'aquests indicadors que tenen la funció de, per exemple, mesurar el nivell de servei, realitzar diagnòstics, comunicar i informar sobre la situació, motivar els equips responsables, etc. En resum aportar una avaluació de qualsevol procés que pugui condicionar la resta d'informació obtinguda d'altres mòduls.



Figura 29: Representació gràfica de possible indicadors KPI

Seguint el mateix cas que en les eines d'interfície gràfica, aquesta capa també permet la incorporació de nous mòduls de dades.

2.2.3. Visió dinàmica dels elements arquitectònics

Una vegada s'ha descrit l'arquitectura des d'una visió estàtica, on s'exposen tots els subsistemes agrupats en el conjunt global del sistema i també des d'una visió funcional, on s'han especificat els diferents components de l'estructura amb una delimitació de les corresponents característiques i funcions particulars, cal contemplar el sistema des d'un punt de vista més dinàmic, entenent quin és el funcionament general resultant de les diferents interrelacions entre els mòduls.

Tot seguit es mostrarà i es descriurà d'una forma concisa, com es comporten aquests components al llarg del temps d'execució i com interactuen entre si, tenint en compte aspectes com el format de les dades d'entrada, les dades de sortida, el tipus de processament de dades, el tipus de connexió, hardware enllaçat, etc. per tal de representar en quines situacions pot esdevenir. Per aconseguir això s'utilitzarà un model que permet descriure l'arquitectura des de diferents vistes dels diferents interessats. Les vistes són:

Vista lògica: La vista lògica està enfocada a descriure l'estructura i funcionalitat del sistema. Es solen utilitzar diagrames de classe, diagrama de comunicació o diagrames de seqüència. Per entendre el projecte des d'una visió general del seu funcionament, el següent diagrama mostra les diferents connexions entre elements de l'aplicació i els recursos utilitzats en cadascun d'ells, especificant les direccions dels processos possibles d'operació:

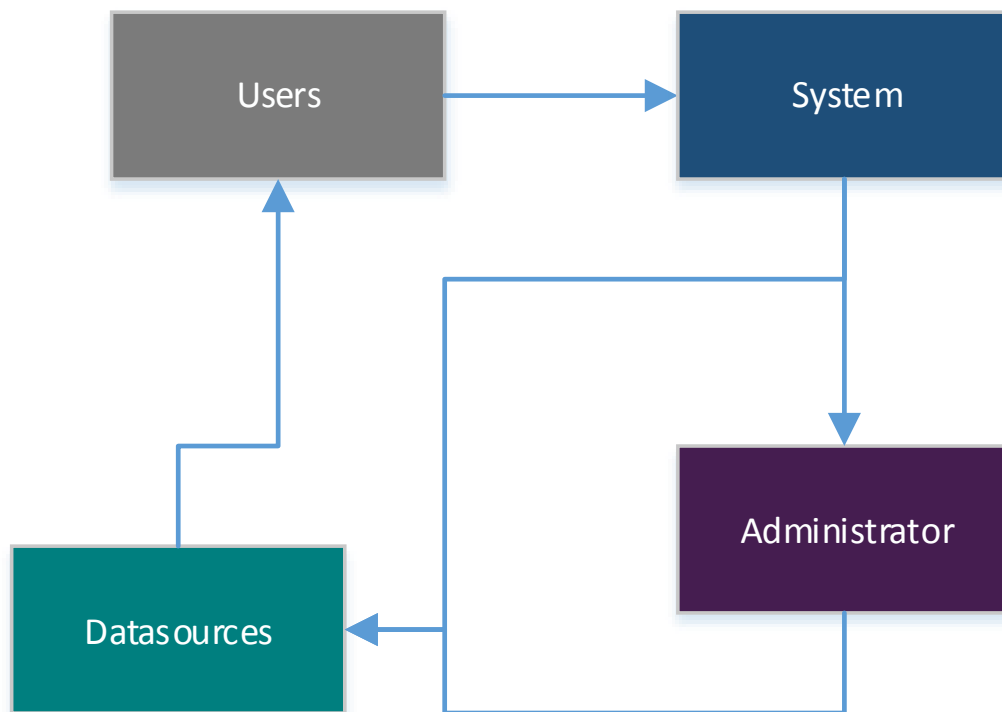


Figura 30: Esquema de la interacció entre els diferents agents involucrats

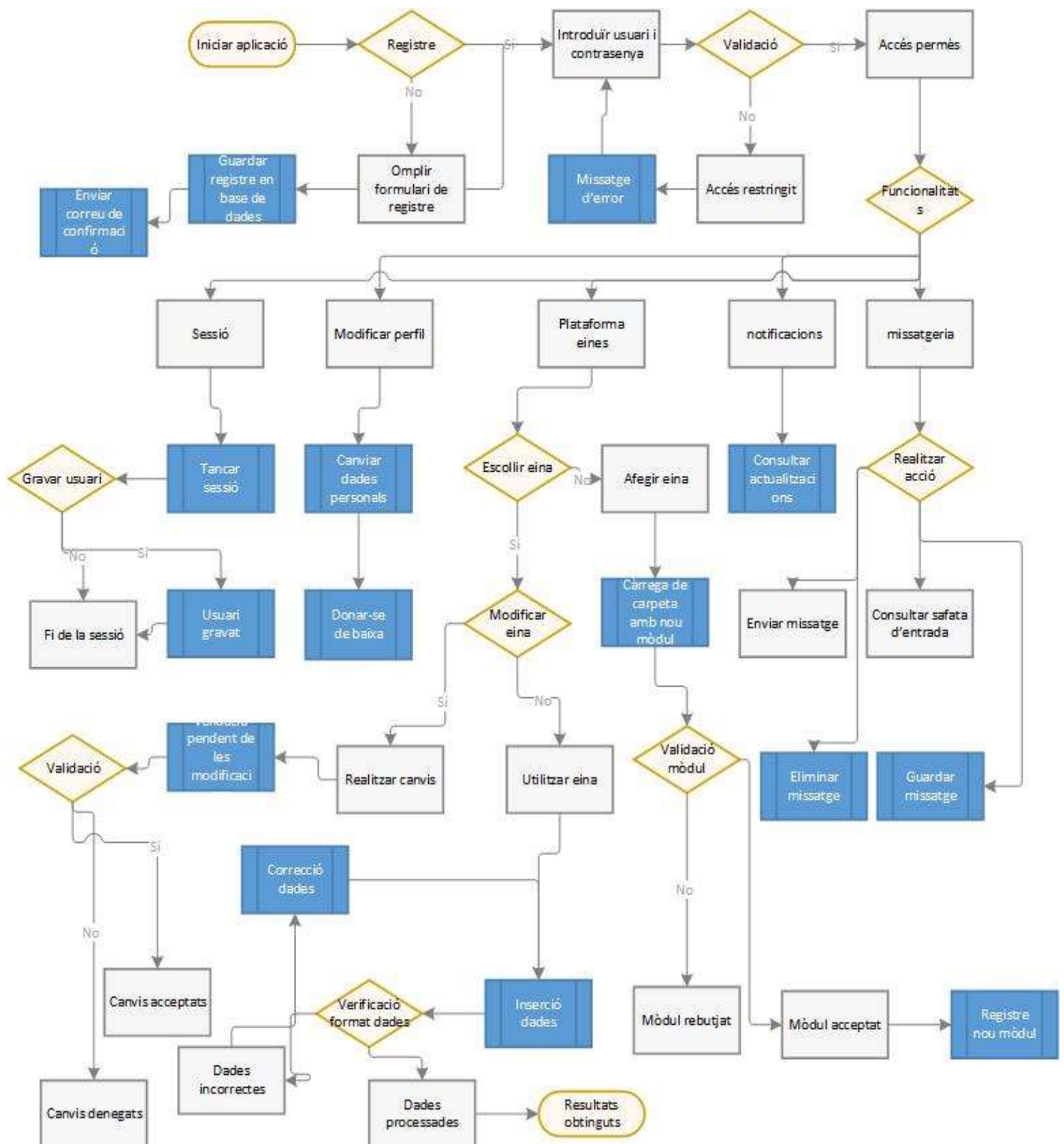


Figura 31: Diagrama procedimental de les relacions entre els diferents casos d'ús i accions

Vista de desenvolupament: La vista des d'una perspectiva del programador i està enfocat en l'administració dels elements de software. Utilitza el diagrama de components UML per descriure els components del sistema.

En aquest diagrama es mostren les diferents pantalles que ha de incloure l'aplicació i quines d'aquestes estan enllaçades entre sí. Com es pot observar l'aplicació estarà composta per una varietat de pantalles, però en la majoria d'elles s'hi accedeix a partir de la pantalla principal. A cada pantalla s'hi detallen els requisits i característiques particulars.

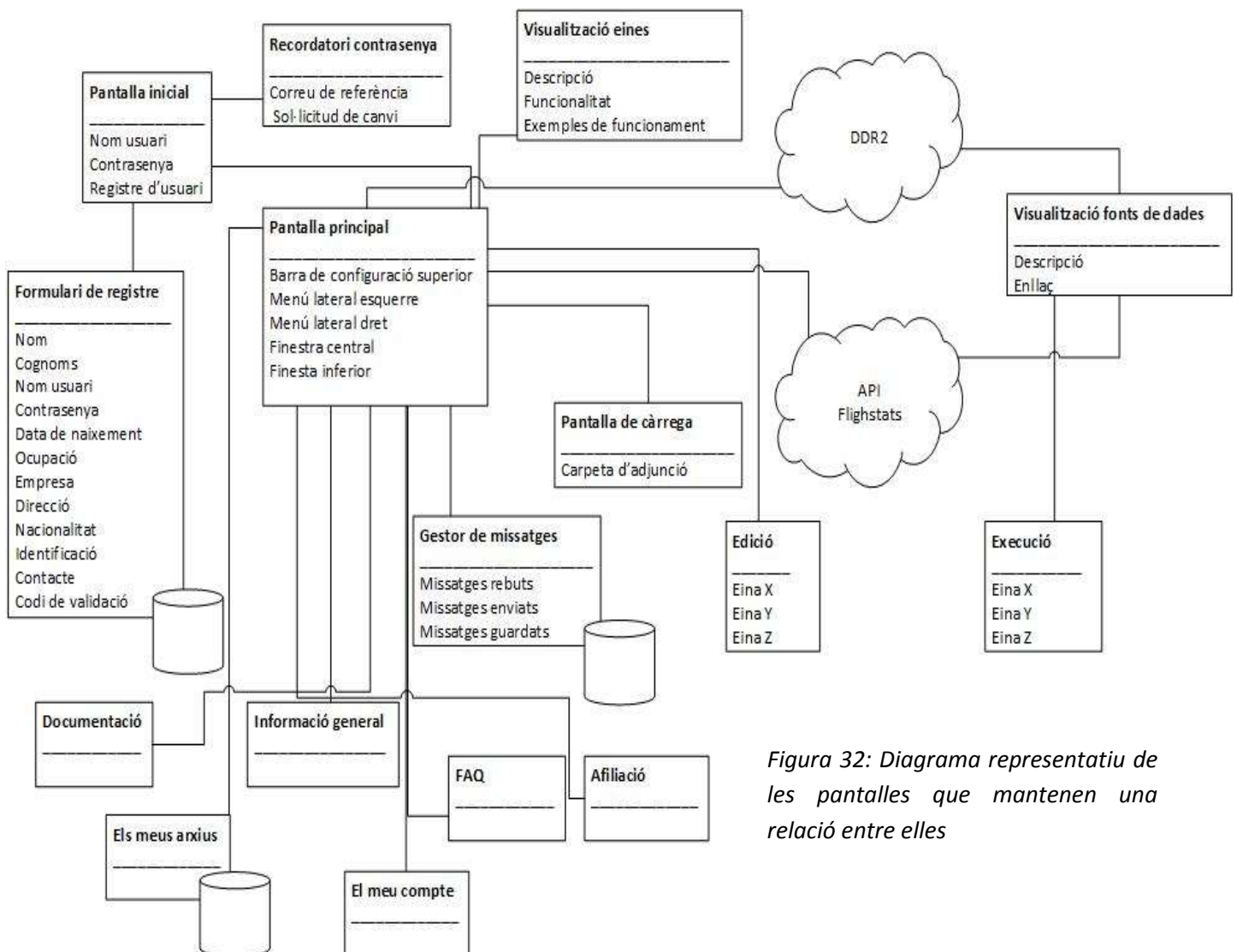


Figura 32: Diagrama representatiu de les pantalles que mantenen una relació entre elles

Vista de procés: La vista de procés tracta els aspectes dinàmics del sistema, explica els processos de sistema i com es comuniquen en temps d'execució. La vista considera aspectes de concurrència, distribució, rendiment, escalabilitat, etc. Es sol utilitzar el diagrama d'activitat.

En aquesta vista es representa el seguiment procedimental per interactuar amb l'aplicació, detallant resumidament les diferents accions i atributs que té cada fase classe.

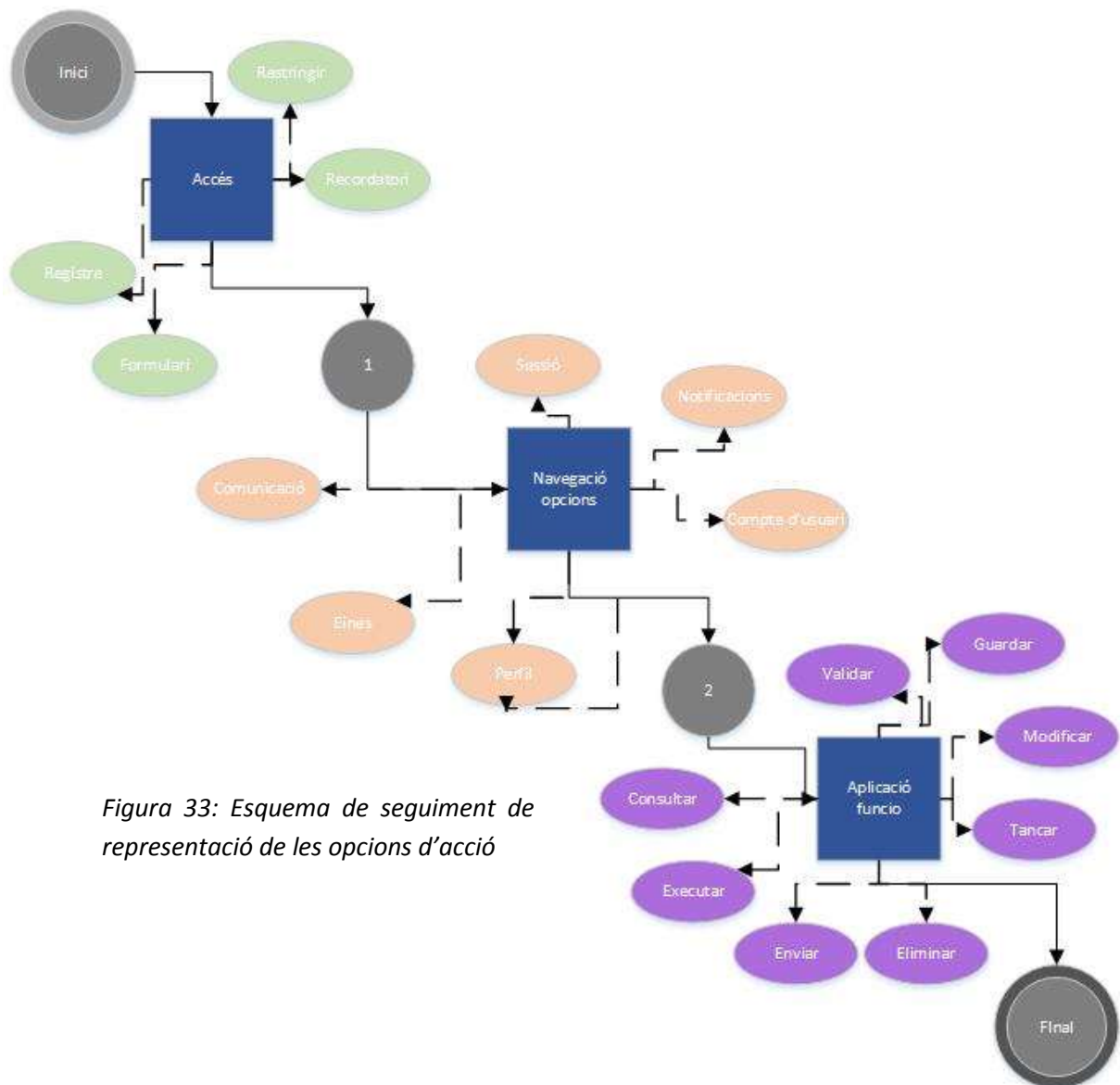


Figura 33: Esquema de seguiment de representació de les opcions d'acció

Vista de desplegament: punt de vista del enginyer de sistemes. Està relacionada amb l'administració de components de software a la capa física, així com les connexions físiques entre aquests components. S'utilitza el diagrama de desplegament.

Com es pot observar en la imatge representada, es segueix una estructura, com ja s'ha dit anteriorment de divisió en capes, en aquest cas s'afegeix una característica més corresponent a les fonts externes que proveeixen la informació.

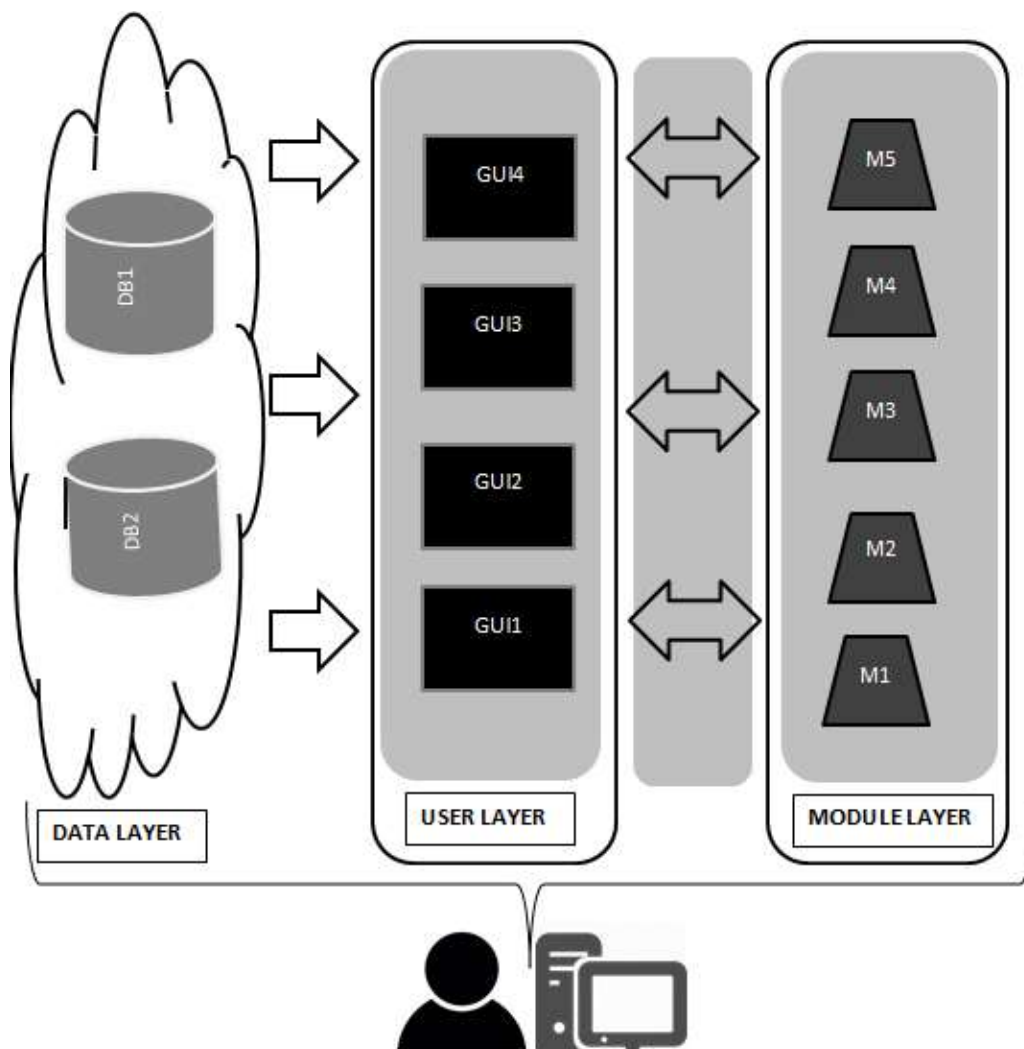


Figura 34: Il·lustració animada del funcionament i estructura de l'aplicació

Les vàries fletxes representades indiquen les direccions dels fluxos existents a través de les diferents parts de la plataforma. en el primer cas s'indica l'existència d'una connexió amb els proveïdors de dades, ja que és requereix tenir accés a fonts de trajectòries i altres formats de dades.

A cada capa s'hi observen els elements que contenen: En el interior de la *user layer* es numeren les possibles eines incorporades al funcionament de la plataforma, en aquest cas n'hi ha quatre, però aquesta capa té la capacitat de poder adoptar qualsevol altra interfície gràfica que sigui necessària. En el cas de la *module layer* mostra la seqüència dels diferents mòduls que componen el sistema. També es pot apreciar les direccions de connexió que existents entre la *user layer* i la *module layer* (connexions bidireccionals).

Per mostrar la informació d'una manera més acurada i que englobi tots els components de l'aplicació, es finalitzaria aquesta secció amb l'exposició de l'arquitectura general. La figura representa l'arquitectura completa de l'aplicació basada en un patró força estandarditzat de divisió de capes. Cada capa té una sèrie d'elements i conceptes que defineixen la funcionalitat bàsica d'una manera simplificada, per facilitar la implementació de l'aplicació que esta representant.

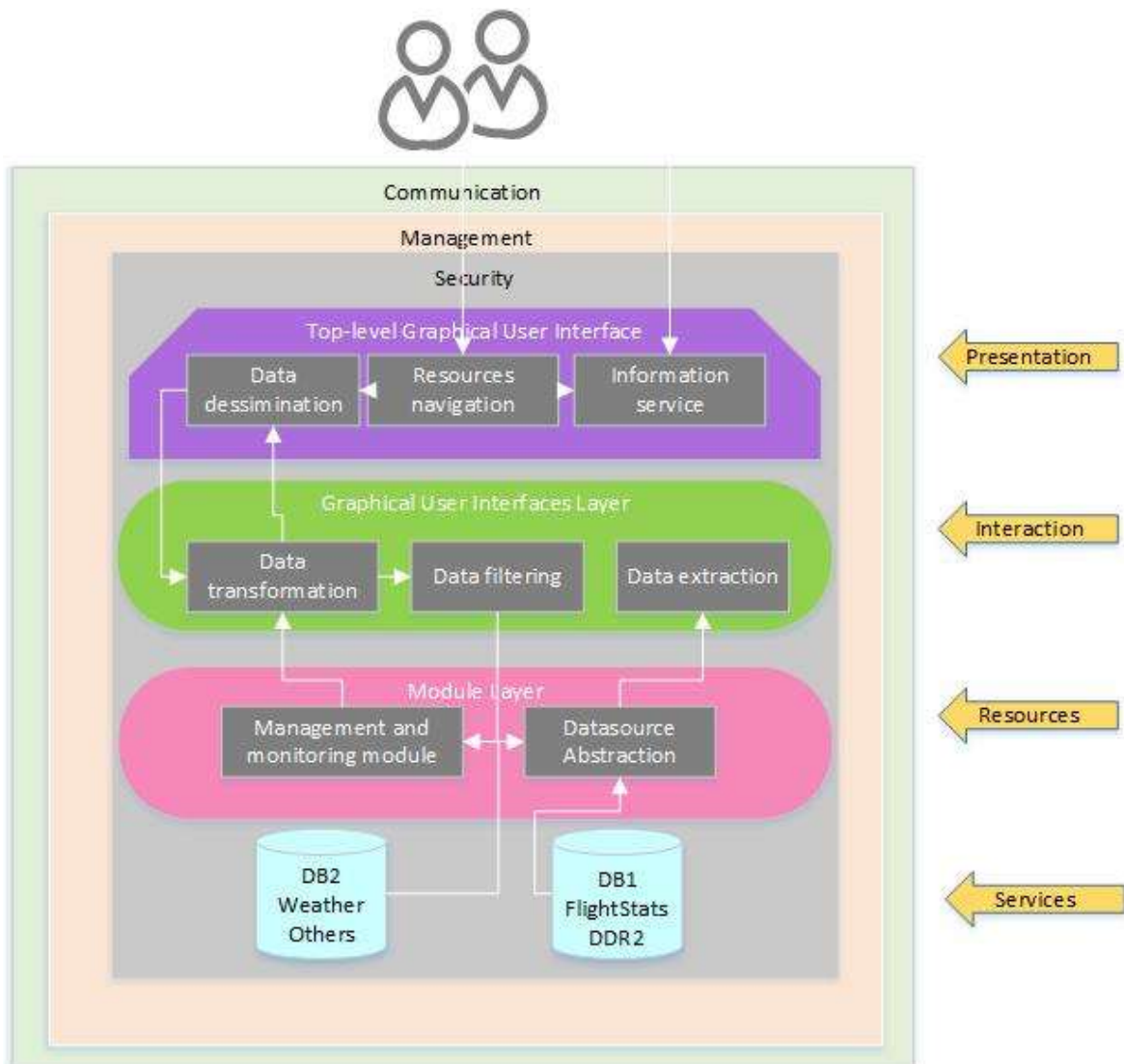


Figura 35: Representació gràfica de les interrelacions existents entre les diferents funcionalitats establertes per mòduls i capes

Per tant es necessari atendre a diferents conceptes que defineixen els requeriments generals sobre el tipus d'informació que una arquitectura ha de mostrar:

Components de la interfície d'usuari: La majoria d'aplicacions necessiten proporcionar d'alguna manera la capacitat de que els usuaris puguin interactuar amb l'aplicació. Els components d'interfície d'usuari faciliten aquesta manera als usuaris per poder-s'hi comunicar, adquirint i validant les dades que vénen d'ells.

Els components de procés: En molts casos, una interacció de l'usuari amb el sistema segueix un procés força predictable. Per ajudar a sincronitzar aquestes interaccions, pot ser útil dirigir el procés utilitzant components de procés separats.

Components de l'activitat. L'aplicació probablement requerirà components que implementen les regles de l'activitat i que duen a terme les tasques de l'activitat. Els components de l'activitat implementen la lògica general de l'aplicació.

Fluxos de treball de l'activitat. Després de definir la informació requerida per un procés d'usuari, les dades poden ser utilitzades per realitzar un procés de l'activitat. els fluxos de treball de l'activitat defineixen i coordinen els processos de l'activitat, de múltiples passos de llarga durada.

Agents de servei. Els agents de servei permet cridar a diversos serveis des de l'aplicació d'una manera monòtona, i poden proporcionar serveis addicionals.

Interfícies de servei. Per exposar la lògica de l'activitat com un servei, s'han de crear interfícies de serveis que donen suport a les capes de comunicació que els usuaris requereixen. interfícies de servei es podrien assimilar a les façanes de l'activitat.

Components lògics d'accés a la informació. La majoria de les aplicacions i serveis necessiten accedir a un magatzem de dades en algun moment durant un procés de l'activitat. D'aquesta manera se centralitza en la funcionalitat d'accés a dades i fa que sigui més fàcil de configurar i mantenir.

Components de l'entitat de l'activitat: La majoria de les aplicacions requereixen el pas de dades entre els components. Les dades s'utilitzen per representar entitats de l'activitat del món real, com ara productes o comandes.

Components per a la seguretat, la gestió i la comunicació: L'aplicació probablement també utilitzarà components per dur a terme la gestió d'excepcions, per autoritzar als usuaris per realitzar certes tasques, i per comunicar-se amb altres serveis i aplicacions.

Com es mostra a la figura, existeix una interfície gràfica (*top-level graphical user interface*) que es troba entre les fonts d'informació i el contingut de l'aplicació, i realitza la funció de permetre al usuari interactuar amb l'aplicació. Aquesta capa simplifica les utilitats de la plataforma de tal manera que es puguin facilitar molts dels processos. En el següent apartat es mostra més detalladament la descripció d'aquesta interfície.

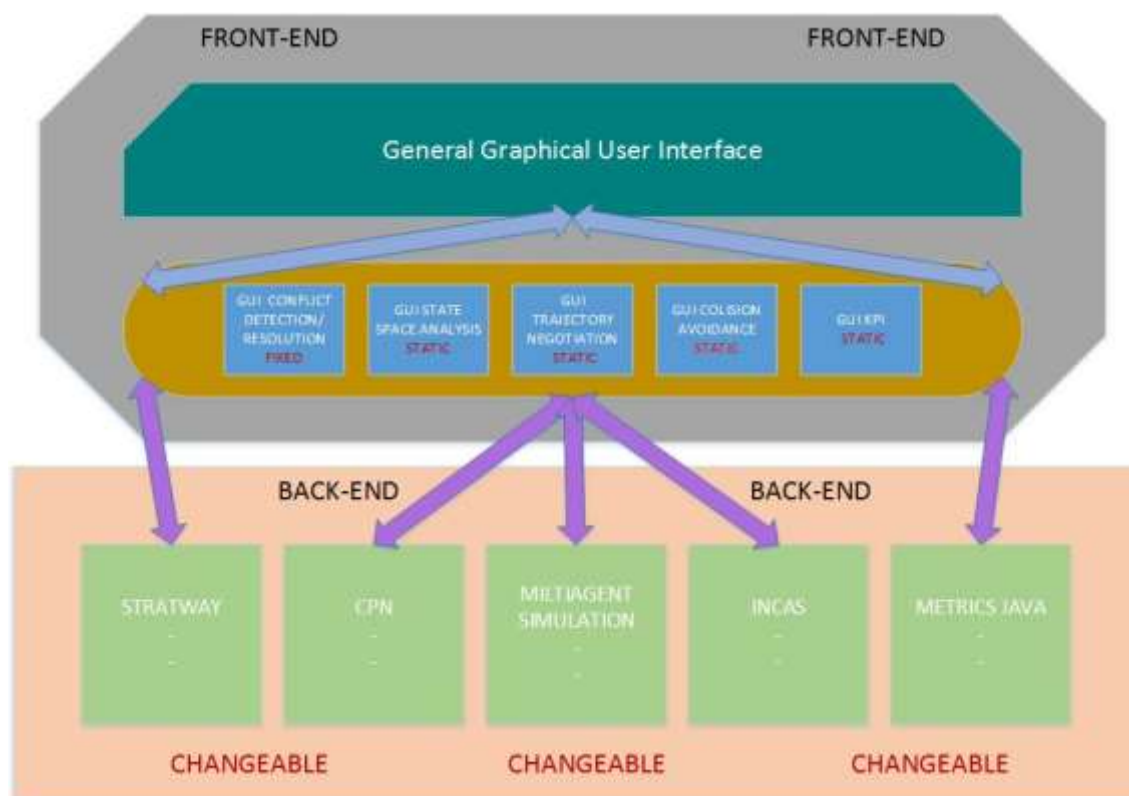


Figura 36: Representació gràfica final dels components de l'aplicació

Com es està representat en la imatge, existeix un primer conjunt corresponent a la part de *front-end* (interacció amb l'usuari), on s'hi inclouen: primer, una capa inicial en representació de la interfície gràfica de l'aplicació més

superficial, i seguidament, es mostra una capa que també esta composta per diferents interfícies gràfiques individuals i fixes per cadascun dels mòduls. A continuació s'observa la part de *back-end* (contingut) on es defineixen les funcionalitats disponibles de l'aplicació.

Per entendre més a fons el concepte es dividirà la figura en els diferents components:

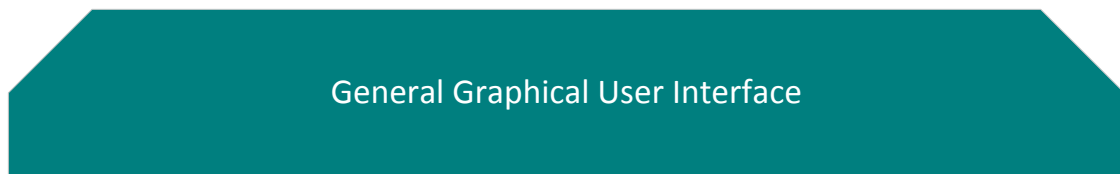


Figura 37: Representació de la capa de la interfície gràfica d'usuari general

En aquesta primera capa es permetrà a l'usuari tenir una visió general de totes les prestacions de l'aplicació, quines opcions i facilitats disposa. Serà la via que tindrà per interaccionar directament amb la resta d'interfícies i que permetrà tenir un coneixement complet de les utilitats i recursos que es poden aconseguir a través dels serveis establerts. Més endavant es realitzarà una descripció gràfica més detallada dels elements que la componen.

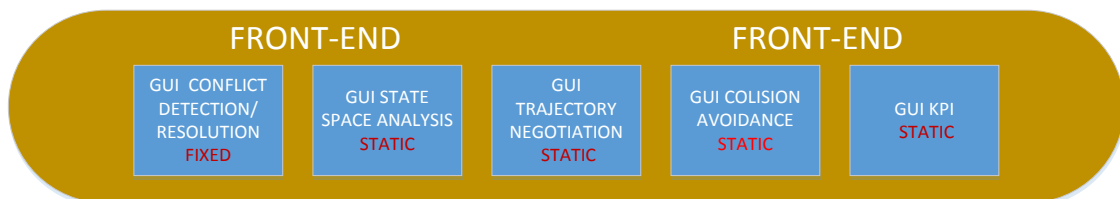


Figura 38: Representació de la capa de les interfícies distribuïdes

Aquesta segona es compon de les diferents interfícies gràfiques individuals per cada mòdul, permetent la utilització, modificació o inserció d'algorismes o altres programes. És la via gràfica de connexió amb el propi contingut de l'aplicació. En aquesta capa les interfícies han de ser fixes degut a que representen la imatge de cadascun dels mòduls, per tant la incorporació o eliminació de mòduls significarà la incorporació o eliminació de les seves respectives interfícies gràfiques d'usuari, a més a més, això provoca una independència

Inter-modular per donar llibertat d'elecció a l'usuari en funció de quin mòdul satisfà millor els seus interessos i necessitats. Això no significa que els mòduls no puguin complementar-se. De fet, l'aplicació també va dirigida a oferir una visió que faciliti una utilització procedimental dels diferents recursos: el primer mòdul es tracta de *Conflict Detection & Conflict Resolution* en primera instància, seguidament en el mòdul de *State Space Analysis* realitzaria un anàlisi dels possibles conflictes no resolts o com a verificació de la resolució anterior s'aplicaria novament una resolució en contemplació de tots els possibles escenaris. A continuació el mòdul de TCAS es realitzaria una última detecció en cas que no hagi pogut determinar una solució factible per cap dels dos anteriors mètodes, realitzant un resolució immediata per tal de evitar possibles col·lisions. Els mòduls de *Trajectory Negotiation* també permet comunicar-se entre els diferents agents per tal de realitzar les modificacions idònies a la situació dels respectius plans de vol. Finalment el mòdul de mètriques KPI es pot aplicar a qualsevol fase d'aquesta procés per tal d'afegir nova informació com ara estimacions estadístiques o de progressió lineal en base els resultats que se'n obtenen. Aquests representen els mòduls inicials ja que són els diferents conceptes i sistemes que s'han considerat bàsics, però existeix la possibilitat d'incorporar futurs mòduls.

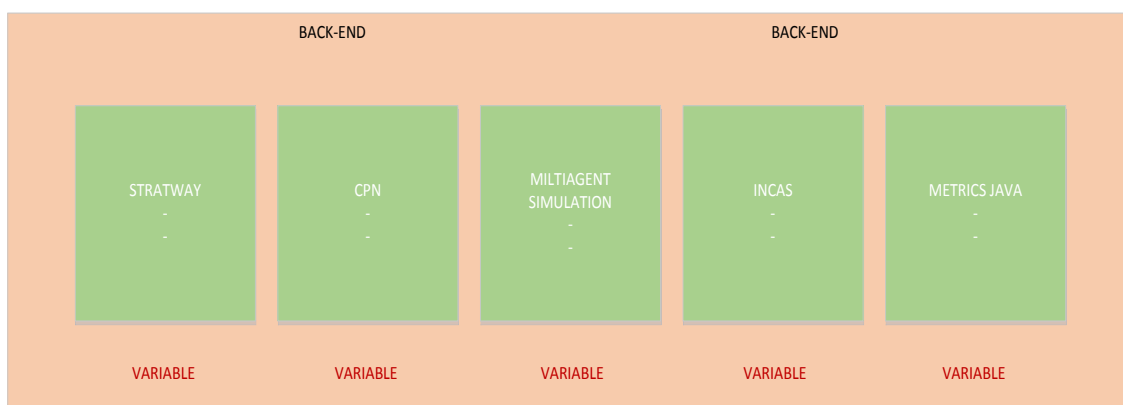


Figura 39: Representació de l'última capa modular (back-end)

Finalment, fora del conjunt de *front-end* existeix un conjunt de *back-end*, on s'inclou tot el contingut útil. Cadascun dels mòduls anteriorment descrits estan connectats amb els seus corresponents apartats d'aquesta capa. A diferència de l'anterior capa, en aquest cas el contingut és totalment variable i modificable: les eines descrites com ara *Stratway*, CPN o InCAS són eines que apareixerien per defecte, degut a que s'ha comprovat i validat prèviament el seu ús, però això no limita la substitució o extensió a noves eines. Aquesta implementació es podria realitzar a través d'una pujada d'arxiu del format en que ha estat creat (per exemple en format *.jar*), o l'altre opció (més recomanable tractant-se d'una aplicació web) seria carregar les eines a incorporar a través del mateix servidor web.

Degut a la importància que té aquesta part de l'aplicació, ja que representa el cos principal, és necessari detallar més a fons cadascun dels paquets per tal de determinar tan característiques generals com específiques (condicions d'entrada o de sortida d'informació):



El primer paquet correspon al mòdul de *Conflict Detection & Conflict Resolution*. En aquest bloc es defineix l'algorisme *Stratway* com a eina per defecte, però permet la incorporació de noves eines destinades a la detecció i resolució de conflictes. Les condicions d'entrada de nous algorismes i programes en aquest paquet són les següents:

Figura 40: símbol primer paquet modular

- Les dades d'entrada han de tenir un format d'arxiu compatible (p.e. .csv).
- La informació introduïda s'ha de constituir com un pla de vol i altres fitxers referents a condicions que puguin afectar als plans de vol (informes de la meteorologia, etc.).
- En els plans de vol es necessita que s'inclogui informació referent als punts cardinals (+ temps) dels diferents *waypoints* d'una trajectòria, nom del vol i identificació de l'aeronau.

Conseqüentment, les dades de sortida han de ser:

- Un nou arxiu on es defineixin les noves trajectòries amb les noves condicions, en funció del procés de resolució aplicat i els paràmetres que hi han afectat.



Figura 41: símbol segon paquet modular

El segon paquet correspon al mòdul de *State Space Analysis*. L'eina per defecte en aquest cas és *Coloured Petri Nets*, però també va destinat a la introducció d'eines relacionades amb una segona resolució de conflictes però en aquest cas considerant diferents escenaris.

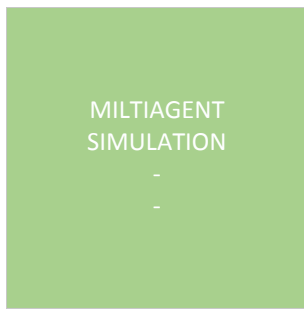
Les condicions d'entrada de nous algorismes i programes en aquest paquet són molt semblants a l'anterior:

- Les dades d'entrada han de tenir un format d'arxiu compatible (p.e. .csv).
- La informació introduïda s'ha de constituir com un conjunt de trajectòries.

- En els plans de vol es necessita que s'inclogui informació referent als punts cardinals (+ temps) dels diferents *waypoints* d'una trajectòria, nom del vol i identificació de l'aeronau.

Conseqüentment, les dades de sortida han de ser:

- Informació en forma de graf, on s'hi representi, a través de nodes i arcs, tots els possibles estats en els que puguin esdevindre els diferents tràfics: estat fora de conflicte, estat amb amenaça de conflicte o estat de col·lisió iminent.



El tercer paquet correspon al mòdul de *Trajectory Negotiation*. L'eina per defecte en aquest cas és *multi-agent simulation*, però també permet afegir altres sistemes de comunicació i de negociació de trajectòries.

Figura 42: símbol tercer paquet modular

Les condicions d'entrada de nous algorismes i programes en aquest paquet són les següents:

- Tipus de comunicació
- Número d'agents que permet interconnectar
- Recursos addicionals
- Tecnologies i equips necessaris per mantenir la comunicació
- Format dels missatges retransmesos

Conseqüentment, les dades de sortida han de ser:

- Avisos i missatges gravats en el format indicat, amb la informació de contingut codificada correctament per tal d'assegurar una seguretat.



Figura 43: símbol quart paquet modular

El quart paquet correspon al mòdul *Col·lision Avoidance*. Aquest mòdul està basat en el sistema TCAS. L'eina per defecte en aquest cas és InCAS, ja que permet la validació d'escenaris on els diferents tràfics involucrats es troben en una situació emergent d'evasió de col·lisió.

Les condicions d'entrada de nous programes en aquest paquet són les següents:

- Importació d'un arxiu que contingui informació referent a l'última fase dels tràfics abans de col·lisió.
- La informació s'ha de classificar en punts cardinals (+ temps) dels diferents *waypoints* d'una trajectòria, nom del vol i identificació de l'aeronau. Com menys separació entre *waypoints*, més precís serà el resultat.

Conseqüentment, les dades de sortida han de ser:

- Una validació de les trajectòries prèviament tractades, i en cas negatiu, una resolució instantània, per tal d'evitar una col·lisió.



Figura 44: símbol cinquè paquet modular

El cinquè i últim paquet correspon al mòdul KPI. La funció d'aquest mòdul és la introducció de diferents mètriques i estimacions realitzades sobre els resultats obtinguts en la resta de mòduls. Les mètriques realitzades en Java són les que s'utilitzarien a priori.

Les característiques principals d'aquest mòdul és que no té requisits d'entrada ni de sortida, sinó que serveix com a complementen per aplicar diferents mètodes matemàtics sobre les dades que s'han obtingut en qualsevol dels paquets anteriors. Per tant l'única condició que s'ha de complir per tal d'incorporar nous elements de KPI és que estiguin en un format compatible amb la informació resultant d'altres mòduls.

Una vegada s'ha definit l'arquitectura de l'aplicació, amb tots els seus components i mòduls que la componen, conjuntament amb les característiques específiques i interrelacions de cadascun d'aquests, es pot concloure aquesta primera part d'anàlisi i disseny conceptual de *l'Open Demonstrator*, deixant oberta aquesta etapa a noves modificacions i millores del rendiment i funcionament general.

2.2.4. Interfície gràfica: vista preliminar a través de *mock-ups*

En aquest apartat es busca identificar les pantalles que composaran la interfície gràfica d'usuari del present Open Demonstrator, part del producte a desenvolupar.

Com ja s'ha definit en l'apartat de les tecnologies utilitzades, la part de *front-end* es desenvoluparà a través de les tecnologies HTML i *Javascript*, amb la possibilitat d'incorporar alguna tecnologia complementària i més específica, com CSS. A través d'aquesta interfície es podran realitzar diferents accions com de modificació, eliminació o integració de mòduls.

En base als requisits funcionals, els diagrames exposats i la definició conceptual de l'arquitectura de funcionament, es pretén facilitar una representació ràpida de l'aparença del producte final visible per l'usuari per tal d'identificar possibles millores abans del disseny sencer i posterior construcció.

Les vistes preliminars d'algunes seccions d'exemple, de la interfície gràfica són les següents:

- La pantalla d'inici és la primera interacció usuari-aplicació i es basarà en permetre l'accés al usuaris registrats per mitjà d'una pantalla creada expressament per introduir el nom d'usuari i la contrasenya, i en cas contrari, permetre el registre d'un nou compte d'usuari. També permet la opció de recordar la contrasenya en cas que s'hagi oblidat.

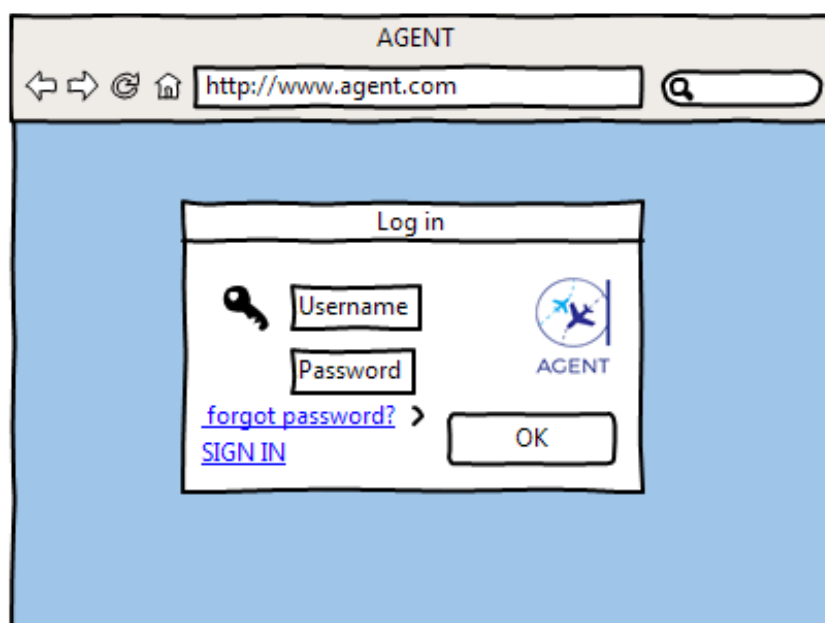


Figura 45: Primera captura de pantalla interfície gràfica

- En cas de necessitat de registre, es redirigirà l'usuari a la següent pantalla per tal que completi el formulari de registre amb les dades personals perquè es puguin emmagatzemar a la base de dades de l'aplicació.

AGENT

http://www.agent.com

SIGN UP

Please fill all the required fields:

Name: Surname:

Address: Gender: ☒ male ☐ female

Company: e-mail:

Username: Password:

Validation Code: Repeated password:

Figura 46: Segona captura de pantalla interfície gràfica

- Un cop que el usuari ja esta registrat i se l'hi ha permès el accés, es donarà pas a la pantalla inicial. Aquesta pantalla inicial correspon a la visualització de la interfície d'usuari gràfica general de l'aplicació (toplevel-GUI). Com es pot contemplar, aquesta pantalla conté diferents elements que permeten realitzar diferents accions, com ara la configuració del compte, d'administració d'arxius o de consulta de questions, però sobretot va dirigida a il·lustrar els diferents mòduls que ofereix.

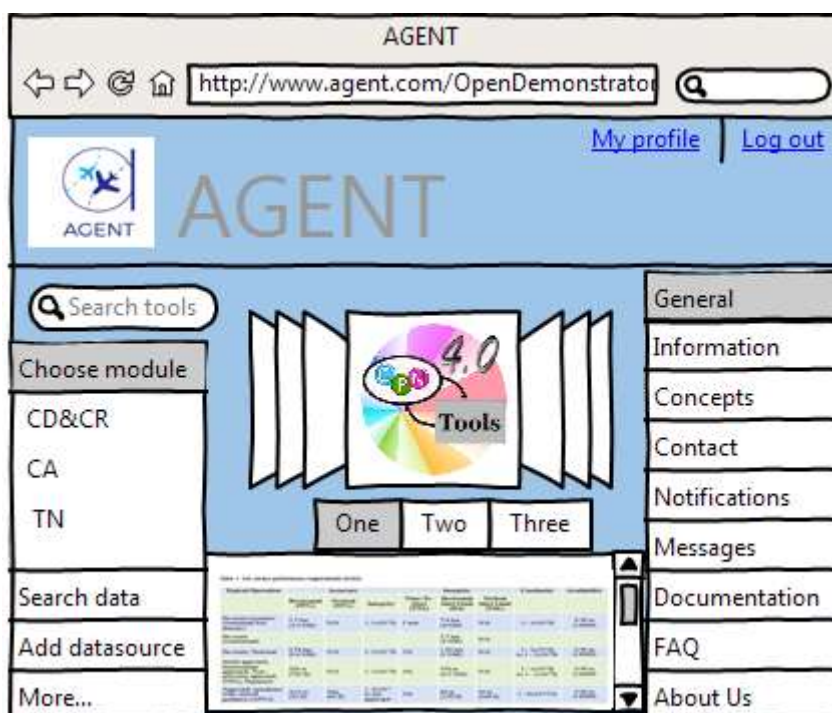


Figura 47: Tercera captura de pantalla interfície gràfica

- Aquesta segona pantalla apareixeria en el moment en que, un cop repassat el catàleg i la visió general de l'aplicació, es decidís utilitzar els serveis que ofereix la aplicació, donant una visió més ampliada de tota la gamma de funcionalitats que es disposa.

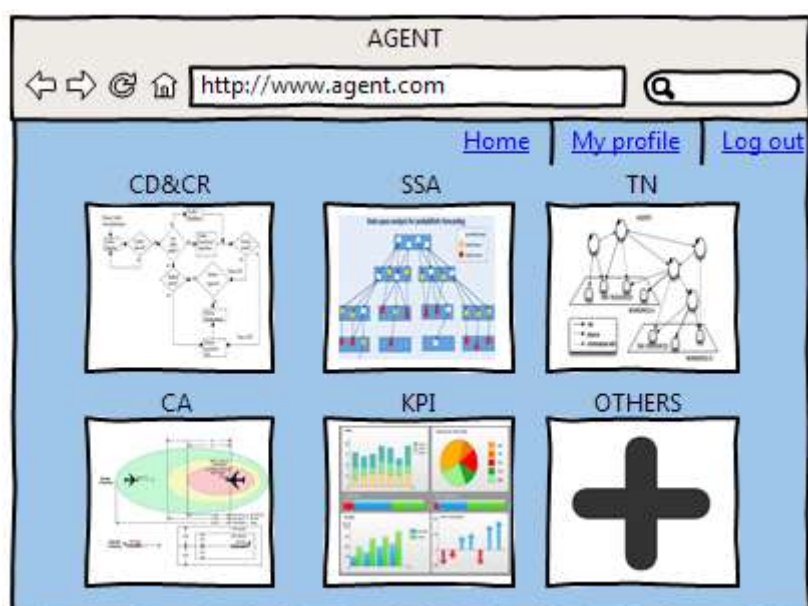


Figura 48: Quarta captura de pantalla interfície gràfica

- A partir d'aquesta interfície gràfica l'usuari accedeix a la pantalla d'interfície gràfica del mòdul escollit. Cadascun del *front-end* particular ha de complir les condicions establertes de cada mòdul. A continuació, es pretén il·lustrar dos exemples d'interfície gràfica particulars, com a mostra d'estructura, en representació a la resta d'interfícies gràfiques individuals. Els dos exemples exposats corresponen als mòduls *Conflict Detection & Conflict Resolution* i KPI:

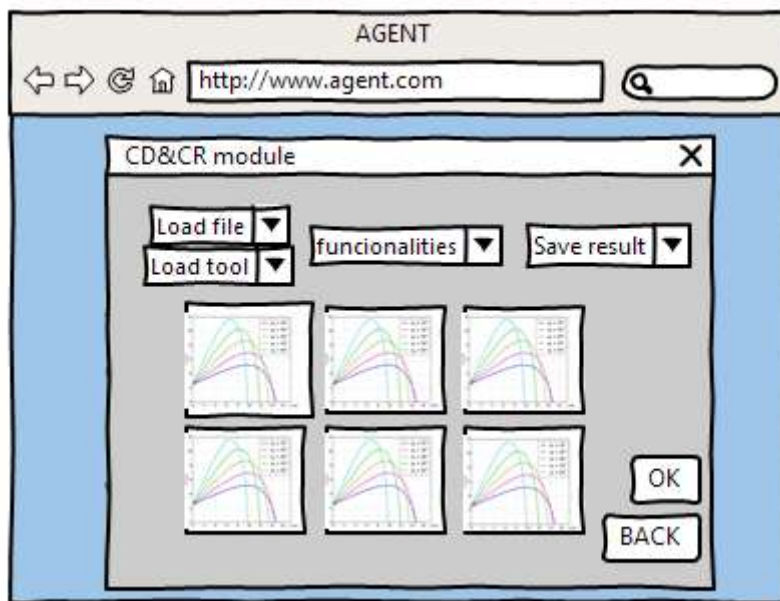


Figura 49: Cinquena captura de pantalla interfície gràfica

En aquest primer exemple es pot observar quin és el procés requerit per tal d'utilitzar o generar noves funcions del mòdul CD&CR. La construcció d'aquesta GUI estaria al voltant de la delimitació de diferents pantalles de representació i diferents pestanyes d'opció.

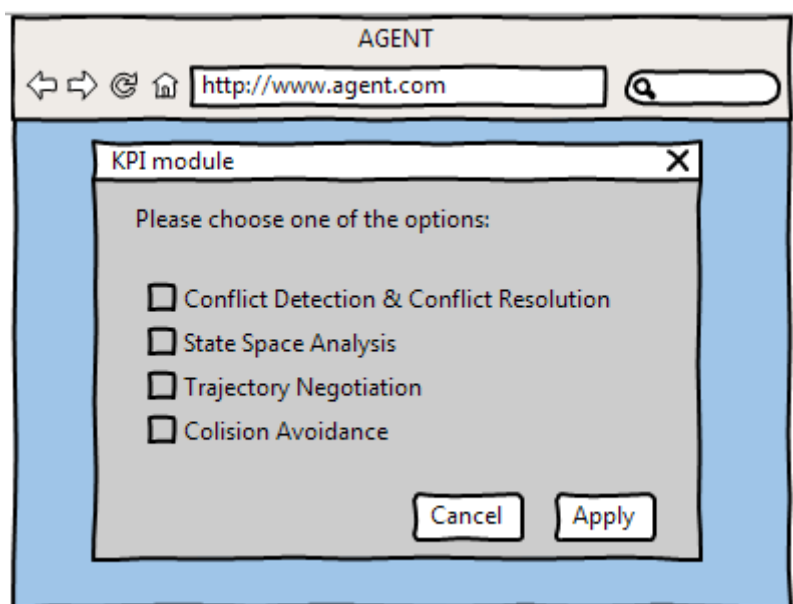


Figura 50: Sisena captura de pantalla interfície gràfica

En aquest segon exemple es pot observar el procés requerit per tal d'utilitzar o generar noves funcions del mòdul KPI. Com prèviament s'ha constatat, aquest mòdul va destinat a la complementació d'altres mòduls, per tant la interfície ha de donar opció d'escollir amb quin mòdul treballarà.

Cal recordar que només es realitza una premissa de la imatge que tindran algunes de les pantalles principals de l'aplicació. De la mateixa manera, aquestes imatges representen una idea inicial que a mesura que avanci el desenvolupament i construcció de l'aplicació, no es descarta que es realitzin les modificacions i adaptacions necessàries. També cal esmentar que no totes les pantalles que la componen estan definides en aquesta tesis.

4. CAS D'ESTUDI: *STRATWAY* - *CONFLICT DETECTION/CONFLICT RESOLUTION TOOL*

En aquest nou capítol es presentarà el cas d'estudi en dos subapartats: descripció del algorisme *Stratway*, atenent a l'entrada de dades, estratègies principals que utilitza per la resolució de conflictes i quin tipus d'informació retorna.

I el segon, on s'aplicarà la metodologia descrita i es procedirà a desenvolupar el cos de la tesi, com a mòdul essencial en la construcció del Open Demonstrator. En aquest subapartat, es detallarà una introducció als mètodes i paràmetres principals que componen aquesta eina i una mostra procedimental del treball pràctic realitzat.

4.1. Coneixement general de l'Algorisme *Stratway*

Aquest apartat va destinat a obtenir una prèvia idea i un coneixement teòric sobre el tema central de la tesi, que és el cas d'estudi de l'algorisme *Stratway*. Aquesta eina ofereix una gran quantitat de funcionalitats, però el funcionament general es descriurà resumidament atenent a: l'entrada de dades, les estratègies principals de resolució que es basen l'algorisme i quin tipus d'informació retorna.

4.1.1. Definició de la funcionalitat

Stratway és un algoritme de resolució estratègica conflictes. Aquest algoritme es centra en el problema d'identificar una ruta planificada d'un avió i fer ajustaments en aquesta ruta per resoldre conflictes amb una altra aeronau o el clima.

Un pla de vol o trajectòria, són les dades que tracta aquesta eina. Hi ha diferents classificacions de trajectòries, però *Stratway* es concentra en les trajectòries 4D, ja que aporten més informació de localització que les altres. Una trajectòria o pla de vol està composta per diferents *waypoints* (punts de coordenades) i *legs* o segments (connexions entre *waypoints*).

Per crear un pla de vol, es pot fer d'una manera manual, creant i afegint diferents *waypoints*, o es pot importar un arxiu de comes separades (.csv) per afegir una o més trajectòries de cop.

NAME	lat	long	altitude	time
Ownship	0.2994	-0.6298	5000.0	0.00
Ownship	0.2994	-0.1579	5000.0	391.36
Ownship	0.2034	0.4806	5000.0	900.42
Ownship	-0.0102	1.1440	5000.0	1800.00
Ownship	0.0663	1.7125	5000.0	2535.49
Traffic	-0.2526	-0.6266	5000.0	0.00
Traffic	-0.2364	-0.1876	5000.0	364.29
Traffic	-0.1040	0.3438	5000.0	812.88
Traffic	0.3402	0.8284	5000.0	1303.99
Traffic	0.9189	1.2190	5000.0	1794.31

Figura 51: Exemple format d'arxiu csv

En aquesta imatge es pot observar la importació de dos plans de vols diferents inclosos en un mateix arxiu .csv, on es mostren els corresponents *waypoints* en funció de la latitud, longitud, altitud i temps.

4.1.2. Aplicació de les estratègies de resolució de conflictes

A continuació es mostren les quatre estratègies bàsiques del programa *Stratway* que resolen conflictes. Cada una d'aquestes estratègies utilitzen tècniques iteratives per buscar solucions:

L'estratègia **Track** busca resoldre al connectar-se a través d'una seqüència de voltes sense canviar la velocitat vertical actual de l'aeronau.

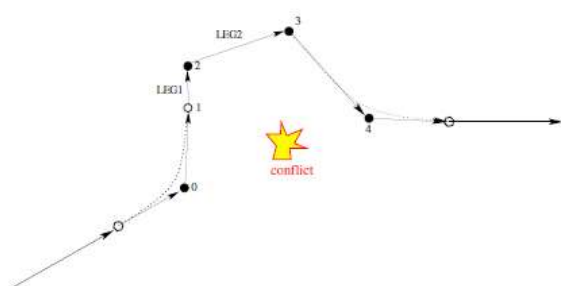


Figura 52: Representació estratègia Track

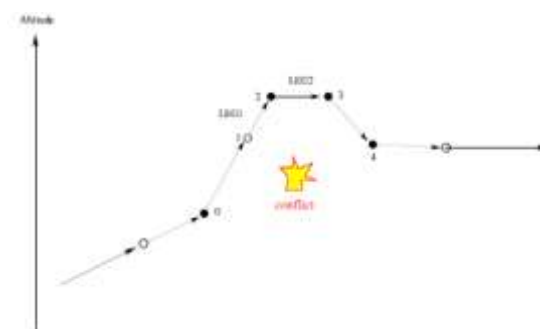


Figura 53: Representació estratègia Vertical

L'estratègia **Velocity** tracta de resoldre els conflictes a través d'una seqüència de canvis de velocitat d'avanç sense canviar la velocitat vertical o d'haver de realitzar voltes.

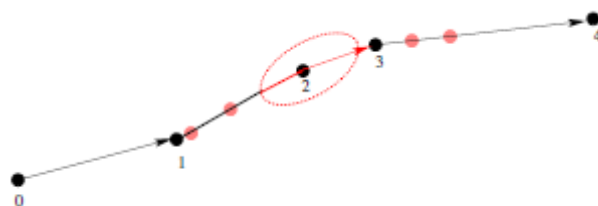


Figura 54: Representació estratègia Velocity



Figura 55: Representació estratègia SideStep.

4.1.3. Informació de sortida

Una vegada s'han aplicat les diferents estratègies de resolució al pla de vol, i la resolució de conflictes ha estat un èxit, es retornarà una nova trajectòria (*kinematic plan*) que contingui els nous *waypoints* lliures de conflictes.

4.2. Utilització de la Java *Application Programming Interface* de *Stratway*

L'API *Stratway* permet que un altre programa pugui cridar funcions i eines necessàries perquè operin sobre un conjunt de plans de vol o trajectòries. Bàsicament la API està basada en les classes principals: *Plan.java*, *Stratway.java* i *Detector.java*.

A continuació es mostra de la forma més bàsica de com utilitzar i cridar l'algorisme *Stratway*. La manera més ràpida per mostrar totes aquestes funcions disponibles, és per mitjà d'un exemple seqüenciat de diferents ordres, que es complementen entre sí, i permet entendre el funcionament global de les tècniques. El exemple està estructurat en tres fases: la importació de informació, la càrrega dels plans a *Stratway* i finalment la comprovació de la resolució dels conflictes.

```
//primera fase
import gov.nasa.larcfm.Util.Position;
import gov.nasa.larcfm.Util.LatLonAlt;
import gov.nasa.larcfm.Util.NavPoint;
import gov.nasa.larcfm.Util.Plan;
import gov.nasa.larcfm.Stratway.Stratway;
import gov.nasa.larcfm.Stratway.Detector;
Stratway stwy=new Stratway();
Plan p1=new Plan();
Plan p2=new Plan();
//segona fase

p1.add(NavPoint.makeLatLonAlt(33.01, -93.00, 35000.0, 1200.0));
p1.add(NavPoint.makeLatLonAlt(33.01, -105.50, 36500.0, 2400.0));
p1.add(NavPoint.makeLatLonAlt(33.01, -87.50, 33000.0, 3800.0));
p2.add(NavPoint.makeLatLonAlt(33.01, -74.50, 36000.0, 2200.0));

p2.add(NavPoint.makeLatLonAlt(33.01, -88.50, 31000.0, 3400.0));
p2.add(NavPoint.makeLatLonAlt(33.01, -104.50, 32000.0, 4200.0));
p2.add(NavPoint.makeLatLonAlt(33.01, -87.80, 37000.0, 2200.0));
p2.add(NavPoint.makeLatLonAlt(33.01, -107.90, 35000.0, 2120.0));

stwy.addPlan(p1);
stwy.addPlan(p2);
```

```
//tercera fase

Plan pf=stwy.resolveConflicts();
Detector d=stwy.getDetector();
If(stwy.getResolutionStatus()==stwy.ResolutionStatusValue.CONFLICT_FREE) {
System.out.println("All conflicts resolved");
}else{
System.out.println("Not all conflicts resolved");
//finally
stwy.getDetector();
}
if (sw.hasError()) { System.out.println(sw.getMessage());
}

stwy.clearPlans();
```

4.2.1. Classes bàsiques

Després d'observar aquest exemple, es pot apreciar diferents parts: construcció del pla principal, lectura i afegiment dels plans de vol, resolució de conflictes, detecció de errors, comprovació del estat del pla de vol, anàlisi dels conflictes i finalment reinici del pla principal. A continuació s'explicaran més detalladament cadascuna d'aquestes parts per entendre millor els petits detalls.

La primera fase del programa és la importació de les diferents classes. Seguidament es procedeix a la creació d'un objecte del tipus *Stratway*.

Stratway.java: La classe *Stratway* és la interfície principal que s'apliquen les estratègies de resolució. Els mètodes més importants , alguns d'ells utilitzats en l'exemple, són:

Stratway():: constructor del objecte.

void clearPlans():: elimina tots els plans inclosos al objecte i neteja tots els missatges relacionats amb les estratègies. No neteja missatges d'errors o avisos.

void addPlan(Plan fp):: afageix un pla de vol al objecte *Stratway*. El primer pla afegit serà el vol subjecte i el següents representaran el tràfic.

`Plan resolveConflicts()`: és el mètode principal per resoldre conflictes. La seva funció és la de processar les diferents trajectòries i retornar un nou pla de vol. Depenen del estat del pla de vol, els conflictes s'hauran resolt o no.

`Plan resolveConflicts(int[] strategies)`: Aquest mètode te la mateixa funció que l'anterior, però en aquest cas et permet definir el ordre d'aplicació d'estratègies.

`ResolutionStatusValue getResolutionStatus()`: retorna el estat de resolució del objecte que s'ha generat al aplicar el mètode `resolveConflicts()`.

`Detector getDetector()`: Retorna un objecte `Detector` que conté informació relativa a l'anàlisi conjunt plans en temps real.

`Detector getDetector(Plan p)`: aquest mètode es comporta semblant al anterior però permet especificar el el pla subjecte.

La segona fase es basa en la construcció dels dos plans de vols existents i la determinació dels diferents *waypoints* que els componen. Un cop s'han definit, s'afegeixen al objecte *Stratway*.

NavPoint.java: Aquesta classe defineix un punt 4-D, que augmenta amb el temps de posició. *NavPoints* es creen generalment amb una posició i un temps.

Plan.java: Un objecte del tipus *Plan* es una seqüència de *NavPoints*. Els plans de vol normalment estant creats a partir d'una pla buit i la seva inserció de diferents *navPoints* utilitzant el mètode `add`. Els mètodes més importants d'aquesta classe són:

`Plan()`: és el constructor d'un pla de vol.

`Plan(String name)`: també és un constructor de pla de vol, permeten l'associació amb un nom.

`int add(NavPoint np)`: aquest mètode et permet afegir un *NavPoint*, el valor retornat és la posició d'aquest *NavPoint*.

`int size()`: retorna la mida del pla de vol. Per recórrer els diferents *waypoints* de la trajectòria, aquesta funció és necessària.

`Plan planFromState(String id, Position pos, Velocity v, double startTime, double endTime)`: retorna un pla de vol nou projectat en la posició i velocitat especificada. També el delimita el temps de començament i d'acabament del pla de vol.

`double getTime(int i)`: retorna els temps(en segons) del punt *i*.

`NavPoint point(int i)`: retorna les coordenades del punt *i* de la trajectòria.

`Position position(double tm)`: retorna la posició en 3D en el temps *tm*.

`Velocity velocity(double tm)`: retorna la velocitat corresponent al temps *tm*.

`Velocity initialVelocity(int i)`: retorna la velocitat al segment de començament *i*.

`Velocity initialVelocity(double t)`: retorna la velocitat al temps *t*.

La tercera fase consisteix en la resolució dels conflictes entre les dues trajectòries. En funció del estat de resolució del nou pla creat, es comprovarà si la resolució ha estat exitosa o hi ha algun error.

ErrorReporter Interface: Totes les classes principals implementen aquesta interfície. Aquesta interfície et permet comprovar els errors i els avisos que s'acumulen en el procés d'execució. Els principals mètodes són:

`hasError()`: Retorna un valor *true* en cas que s'hagi trobat algun error, és a dir, algun tipus de informació inconsistent.

`hasMessage()`: retorna un valor *true* en cas que es trobi algun avís. Els avisos indiquen sucusos inusuals (no necessàriament incorrectes).

`string getMessage()`: retorna una cadena de caràcters que en representació d'algun error o avís.

`string getMessageNoClear()`: té la mateixa funció que l'anterior mètode, però en aquest cas no es neteja el estat de avisos o errors.

Finalment, En cas que no hagi resolt tots els conflictes es pot procedir a l'anàlisi i detecció de problemes d'estratègia, per tal de trobar una solució adequada.

Detector.java: Aquesta classe conté informació detallada sobre tots els conflictes entre el vol subjecte i el tràfic. Els principals mètodes d'aquesta classe són:

`int size():` Retorna el número de conflictes que s'han detectat.

`boolean conflict():` Retorna el valor *true* si es detecta un conflicte.

`double getTimeIn(int i):` Retorna el temps d'inici del conflicte en el punt i.

`double getTimeOut(int i):` Retorna el temps de fi del conflicte en el punt i.

`double getTimeClosest(int i):` Retorna el temps aproximat /a prop del conflicte del punt i.

`getClosestVert(int i):` Retorna la distància vertical entre dos avions en conflicte i.

`getClosestHoriz(int i):` Retorna la distància horitzontal entre dos avions en conflicte i.

`int getTrafficID(int i):` Retorna la identificació del tràfic en el conflicte del punt i.

`void sortConflicts():` classifica els diferents conflictes detectats en funció de temps d'inici.

Existeixen altres classes realment necessàries per el correcte funcionament del programa *Stratway*, com ara les classes que serveixen per definir les posicions dels *4D waypoints* (*Point.java*, *LatLonAlt.java*, *Position.java*, *NavPoint.java*, *Velocity.java*), classes útils per la creació de polígons que fan referència a zones climàtiques que han de ser evitades o classes que permeten el traspàs de una trajectòria lineal a un *kinematic plan* (*TrjGen.java*), etc.

Aquestes classes també són molt importants, però en aquesta tesis, només cal que es coneguin sobre la seva existència d'una manera superficial, ja que no s'hi treballen directament.

4.3. Seguiment del treball realitzat

En aquesta secció es descriuen les diferents funcionalitats que s'han afegit durant la participació en el desenvolupament de l'aplicació. Afegint noves classes i mètodes, per tal que complementin la funció de l'eina *Stratway*. A partir de l'exposició d'un cas pràctic que permeti experimentar algunes d'aquestes funcionalitats adherides, es podrà realitzar un seguiment del processament de dades existent en l'eina.

Es comença amb una descripció de l'organització general del entorn de treball per facilitar la correcta utilització de les diferents classes i llibreries existents:

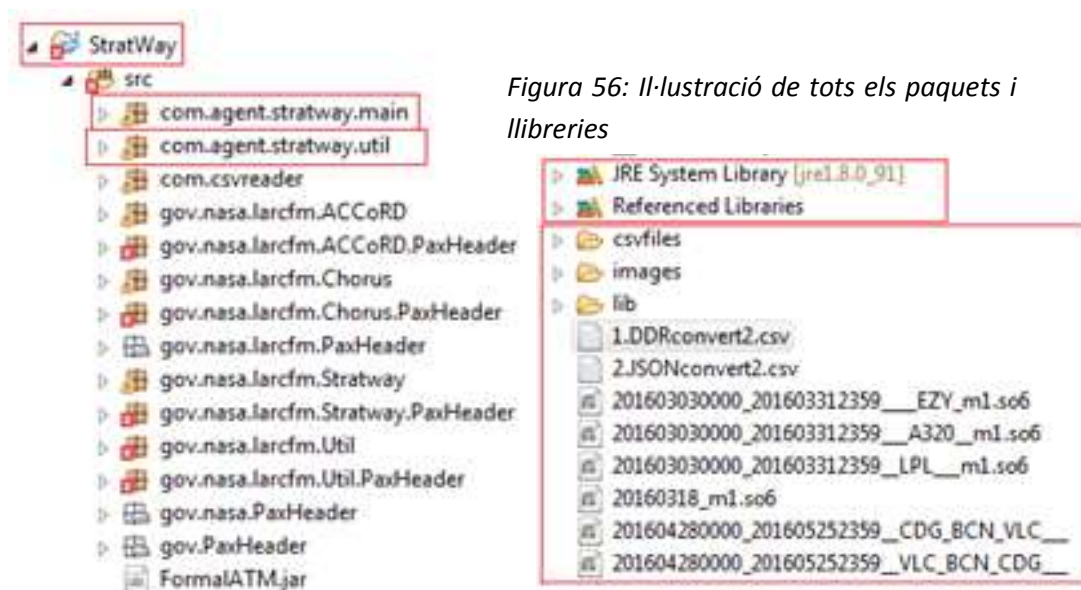


Figura 56: Il·lustració de tots els paquets i llibreries

Com es pot observar existeix una carpeta general del projecte on s'engloben les diferents sub-carpetes creades: En el interior de la carpeta *src* s'identifiquen diferents paquets que contenen multitud de classes interconnectades entre elles. Els paquets més importants que cal conèixer són: 1. *com.agent.stratway.main*, on s'inclou la classe que representa la interfície gràfica de l'algorisme; 2. *com.agent.stratway.util*, on s'inclou les classes amb les quals s'ha treballat en aquesta tesis, 3. *Gov.nasa.larcfm.Stratway*, 4. Secció de les llibreries i 5. Secció de les carpetes i arxius externs.

La resta de paquets també s'utilitzen i són necessaris, però la seva afectació en aquesta tesis es podria classificar com a secundària.

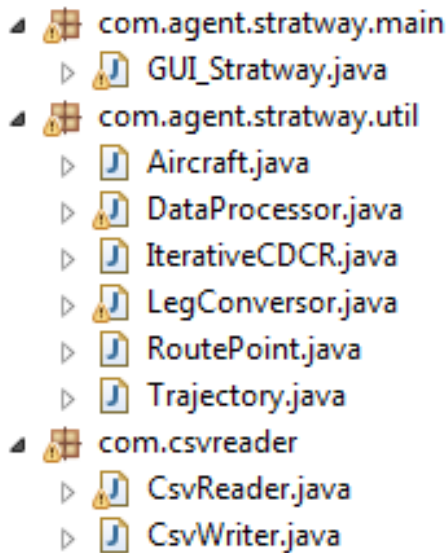


Figura 57: Il·lustració dels paquets i classes principals

En el paquet *com.agent.stratway.main* com indica el seu nom, representa el paquet principal que fa cobrar sentit a la resta de paquets. En ell s'hi troba la classe *GUI_Stratway.java* on es defineixen tots els elements visuals que permetran interactuar l'usuari amb les diferents funcionalitats de l'aplicació.

Seguidament es llisten una sèrie de classes que són objecte principal d'estudi d'aquest apartat del projecte: *DataProcessor.java*, juntament amb *RoutePoint.java* i *Trajectory.java*, s'han creat per permetre obtenir informació per mitjà de l'extracció de dades referents a les trajectòries i condicions de vols. *LegConversor.java*, juntament amb *CsvReader.java* i *CsvWriter.java*, és un conjunt permet una divisió d'aquestes trajectòries en diferents punts en funció dels interessos del usuari. Finalment *Aircraft.java* i *IterativeCDCR.java* són dues classes que corresponen a una futura funcionalitat de permetre una resolució de conflictes considerant un procés iteratiu d'assignació, on es contemplin tots els punts de vista possibles.

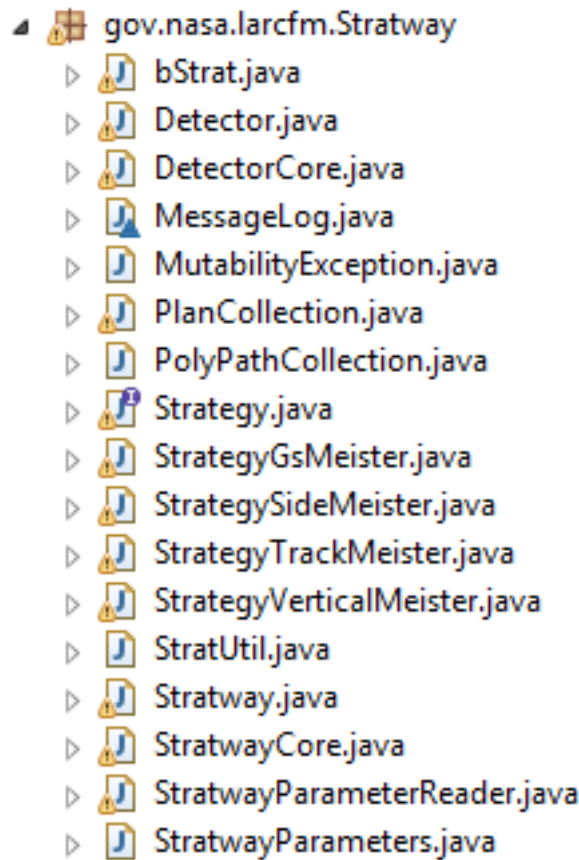


Figura 58: Il·lustració de les classes pròpies de Stratway

En aquest paquet anomenat *gov.nasa.larcfm.Stratway*, representa la part més essencial de l'eina, a les diferents classes que componen el algorisme conegut com a *Stratway*, ja que es pot observar que inclou les classes bàsiques anteriorment descrites. Bàsicament permet la configuració de plans de vol, de creació d'escenaris de resolució de conflictes i de la utilització dels paràmetres i les estratègies principals. Sobre aquest últim ús, se'n parlarà més endavant durant el seguiment del exemple exposat.

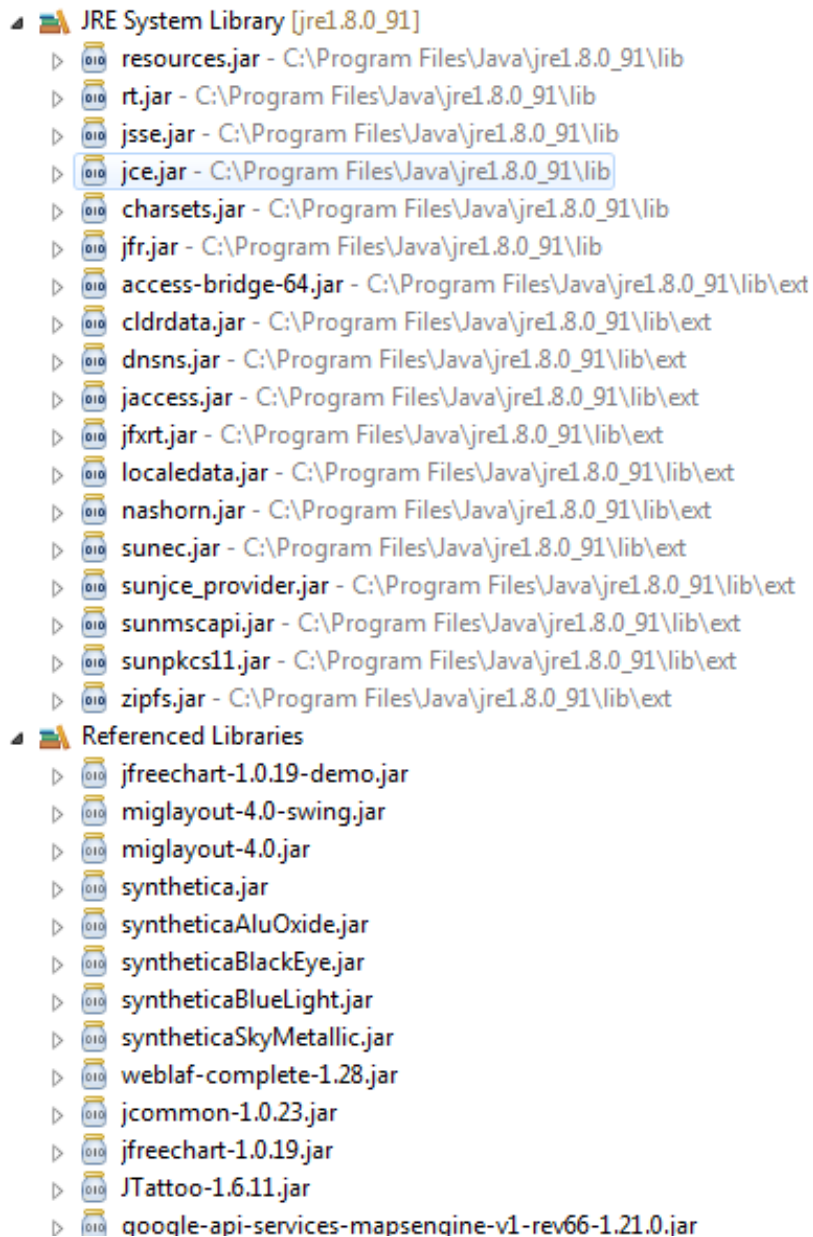


Figura 59: Il·lustració de les llibreries utilitzades

En aquesta imatge es pot observar una gran llista d'arxius .jar com a llibreries del sistema o referenciades.

També es pot visualitzar tots els arxius que s'utilitzen, es modifiquen o es generen durant el procés d'execució d'algunes classes com ara els arxius .csv obtinguts de la conversió JSON.

4.3.1. Interfície *GUI_Stratway.java*

La classe *GUI_Stratway.java* és l'escenari principal d'aquesta eina. En aquesta interfície s'hi recopilen totes les característiques visuals de l'eina, que permeten obtenir un valor retornat de l'algorisme.

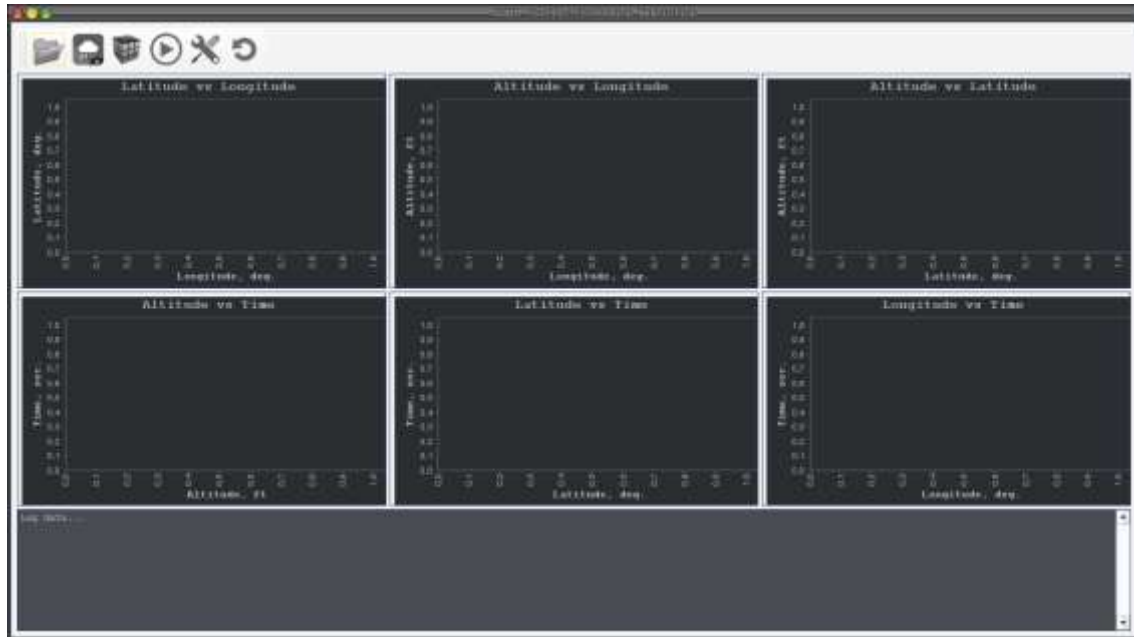


Figura 60: Captura de pantalla de la *GUI_Stratway*

Aquesta captura de pantalla ens mostra la composició de la interfície. A la part de dalt s'observen una barra d'utilitats on s'hi identifiquen diferents botons: seguint un ordre d'esquerre-dreta, el primer botó (símbol: carpeta), té la funció de permetre la càrrega de plans de vol, en un format d'arxiu .csv (mateix format que l'exemple de l'apartat anterior); El segon botó (símbol: núvol) permet la càrrega de dades sobre l'estat meteorològic; El tercer botó (símbol: cub de *rubik*) fa la funció d'aplicar el algorisme de detecció i resolució de conflictes sobre els diferents plans de vol introduïts, tenint en compte els condicionants del clima; el quart botó (símbol: *play*) té la finalitat de representar una animació dels nous plans de vol retornats; El cinquè botó (símbol: eines) permet la re-configuració dels paràmetres que es tenen en

compte a la hora d'aplicar les diferents estratègies de resolució. D'aquesta manera permet una adequació de la resolució a la situació.

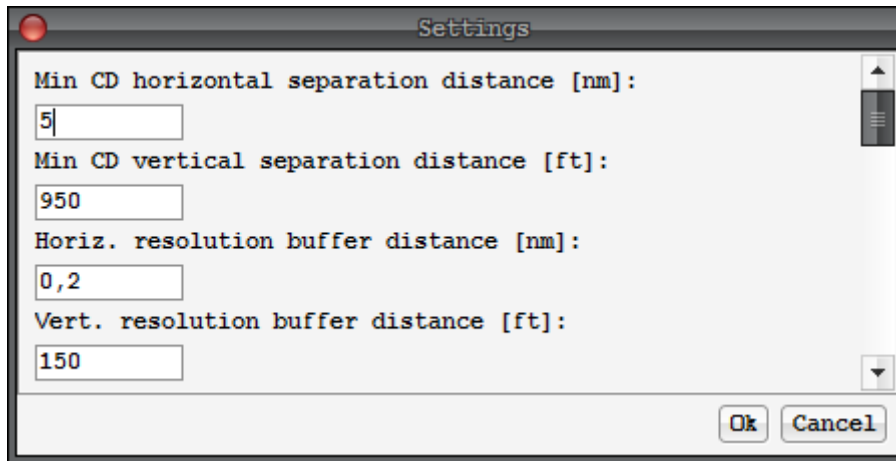


Figura 61: interfície gràfica valors paràmetres

En aquesta imatge es poden observar alguns dels paràmetres que afecten a la definició de les noves separacions entre els diferents vols; finalment el quart botó (símbol: fletxa) s'encarrega de netejar la pantalla i esborrar totes les dades sobre les trajectòries anteriors, permetent crear un nou escenari.

Seguint amb la figura inicial, la següent part de la interfície correspon a les sis gràfiques visuals representades. Aquestes finestres permeten tenir sis diferents perspectives de les trajectòries introduïdes, en funció de les unitats dels eixos de coordenades X i Y. En aquest cas estan definides segons latitud-altitud, longitud-altitud, longitud-latitud, latitud-temps, altitud-temps i longitud-temps. És a dir, diferents combinacions possibles entre les 4 dimensions que componen les trajectòries.

L'última part de la interfície s'hi mostra un panel on es visualitzarà els diferents estats possibles: que s'hi hagi introduït dades o no, i que s'hagi resolt conflictes o no. En el moment que s'introdueixen les trajectòries es mostra informació escrita referent als plans de vol (*waypoints*, nom del vol). En el moment que es pretén resoldre els conflictes, també té la funció de

verificar que hagi estat una resolució exitosa, i en cas contrari, que mostri la informació de la fallida (estratègies, punts, temps).

Abans de començar a fer proves amb aquesta eina, el primer pas és obtenir un arxiu que inclogui les dades necessàries per identificar dues o més trajectòries. Per aconseguir això, com s'ha dit definit en el disseny, es té el suport de dues fonts de dades que proveeixen la informació que es demana, amb la possibilitat d'obtenir-ne a temps pràcticament real.

Flightstats proveeix dades en format JSON. D'aquesta manera es prescindeix de descàrrega, ja que es poden interpretar i utilitzar les dades d'una forma directe.

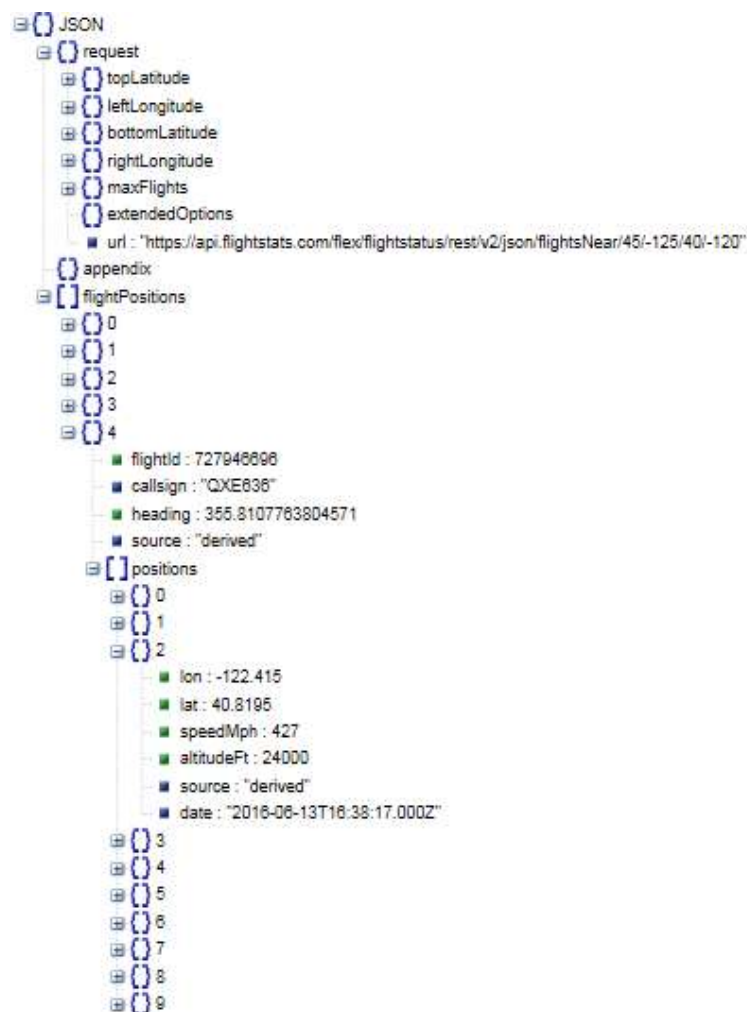


Figura 62: Exemple d'arxiu en format JSON

En aquesta imatge es pot observar un exemple d'obtenció d'informació a través d'aquest mètode. És de gran utilitat ja que atorga la informació essencial per incorporar-la a la eina *Stratway*. Bàsicament les dades que ofereix és la identificació de l'aeronau, el número de vol, velocitat, rumb i les diferents posicions (altitud, latitud, longitud) en funció del temps.

Per realitzar les diferents recerques de vol *flightstats* posa a la disponibilitat un ampli ventall de filtres i cerques diferents d'informació, considerant aeroports de sortida, d'arribada, dates, coordenades o altres. Però per obtenir dades a temps real, les cerques que ho permeten són dues: *Bounding box search*, *Point and distance search* (corresponents al paquet de *FlightNear*).

Flights within bounding box `/v2/json/flightstatsNear/{topLat}/{leftLon}/{bottomLat}/{rightLon}` GET

DESCRIPTION
Get all flights with current positions inside a specified bounding box. Bounding box coordinates are specified in degrees of latitude/longitude. Sides of box must not exceed 5 degrees in length.

PARAMETER	VALUE	DESCRIPTION
appld	12943bf7	Application ID
appKey	0d34393a13924333a19ae71c	Application key
topLat	45	Latitude for top of bounding box
leftLon	-125	Longitude for left side of bounding box
bottomLat	40	Latitude for bottom of bounding box
rightLon	-120	Longitude for right side of bounding box
maxFlights	5	The maximum number of unique flights to return data for. Primarily used to restrict the response size for testing.
extendedOptions		Extended options for modifying standard API behavior to fit special use cases. Options: 'useInlinedReferences', 'useHttpErrors', 'includeDeltas'.

Send Request [HIDE RESPONSE](#)

Figura 63: exemple de cerca a través de bounding box

Bounding box search és un tipus de recerca de vols que es basa en la delimitació d'una àrea en forma de cub sobre l'espai aeri, amb l'objectiu de captar tots aquells vols que es trobin en aquell precís moment en el seu interior. A través de diferents coordenades i número de màxim de vols, aquesta cerca pot ser molt útil.

► **Flights within area specified by point and distance** /v2/json/flightsNear/{lat}/{lon}/{miles} GET

DESCRIPTION
Get flights centered around a particular point, within a given radius. The area searched is a square (not circle); the radius specifies the distance from the center of the square to one of its corners. Center point is specified in degrees of latitude and longitude; radius is specified in miles.

PARAMETER	VALUE	DESCRIPTION
appId	(required)	Application ID
appKey	(required)	Application key
lat	45.00	Latitude for center point
lon	-122.0	Longitude for center point
miles	25	Radius within which to include flights (miles, <= 200)
maxFlights	5	The maximum number of unique flights to return data for. Primarily used to restrict the response size for testing.
extendedOptions		Extended options for modifying standard API behavior to fit special use cases. Options: 'useInlinedReferences', 'useHttpErrors', 'includeDeltas'.

Figura 64: exemple de cerca a través de point and distance

Point and distance search és l'altre opció, i en aquest cas es basa en delimitar un punt geogràfic concret de dues dimensions i una distància (en milles). D'aquesta manera s'aconsegueix crear una circumferència amb un determinat diàmetre, que englobi una quantitat determinada de vols.

La segona font d'informació, es tracta de DDR2. La figura següent ens mostra un exemple d'arxiu proveït per DDR2.

```

EPKH_EPKH EPKH EPBY AN28 075000 082000 8 8 2 PLF112 131220 131220 3004.666667 1187.083333 3004.666667 1187.083333 173057969 1 0.000000 0
UKHH_UKHH UKHH UKHH 2222 074000 124000 5 5 2 URRUB 131220 131220 2995.616667 2177.400000 2995.616667 2177.400000 1500163354 1 0.000000 0
ENUN_ENNJ ENUN ENNJ AS32 084000 084046 1 1 2 HXS60R 131220 131220 3856.300000 435.083333 3856.233333 431.983333 173059808 1 1.347402 0
EHGG_EHGG EHGG EHTE BE58 131246 131300 0 0 2 PHETA 131220 131220 3187.500000 395.000000 3187.500000 395.000000 173065546 1 0.000051 0
ENNJ_ENUN ENNJ ENUN AS32 082500 082546 1 1 2 HXS60R 131220 131220 3856.233333 431.983333 3856.300000 435.083333 173059380 1 1.347402 0
!Kugp_!Kugp 2222 2222 S92 215500 215556 10 10 2 BHL507 131220 131220 3508.000000 111.000000 3506.000000 113.000000 173078673 1 2.257240 0
!KwQc_!KwQc 2222 2222 ENXL S92 194500 194522 10 2 2 BHL207 131220 131220 3382.000000 195.000000 3382.566667 195.866667 173076432 1 0.742616 0
!OUVU_!OUVU 2222 2222 AN26 103500 110604 80 80 2 UNO968 131220 131220 1446.000000 -799.000000 1333.000000 -789.000000 173062512 1 113.373373 0
ERCH_$AACD ERCH EGMX B752 223200 223224 0 9 0 BCS3318 131219 131219 3337.083333 759.366667 3336.550000 758.683333 173052651 1 0.658319 0
$AACD_$AAE ERCH EGMX B752 223224 223252 9 15 0 BCS3318 131219 131219 3336.550000 758.683333 3336.000000 758.000000 173052651 2 0.671943 0
$AAE_$AACF ERCH EGMX B752 223252 223414 15 35 0 BCS3318 131219 131219 3336.000000 758.000000 3334.616667 755.666667 173052651 3 1.911133 0
$AACF_$AAG ERCH EGMX B752 223414 223423 35 40 0 BCS3318 131219 131219 3334.616667 755.666667 3334.283333 755.083333 173052651 4 0.468900 0
$AAG_$AAGS ERCH EGMX B752 223423 223433 40 40 2 BCS3318 131219 131219 3334.283333 755.083333 3333.933333 754.500000 173052651 5 0.480923 0
$AAGS_$AACH ERCH EGMX B752 223433 223442 40 40 2 BCS3318 131219 131219 3333.933333 754.500000 3334.100000 753.633333 173052651 6 0.517617 0
$AACH_$AACI ERCH EGMX B752 223442 223501 40 48 0 BCS3318 131219 131219 3334.100000 753.633333 3334.450000 751.883333 173052651 7 1.049498 0
$AACI_$AACJ ERCH EGMX B752 223501 223506 48 50 0 BCS3318 131219 131219 3334.450000 751.883333 3334.616667 751.000000 173052651 8 0.526444 0
$AACJ_$AACK ERCH EGMX B752 223506 223551 50 70 0 BCS3318 131219 131219 3334.616667 751.000000 3335.483333 746.616667 173052651 9 2.624654 0
$AACK_$AACL ERCH EGMX B752 223551 223558 70 70 2 BCS3318 131219 131219 3335.483333 746.616667 3335.650000 745.750000 173052651 10 0.517312 0
$AACL_$AACH ERCH EGMX B752 223558 223614 70 77 0 BCS3318 131219 131219 3335.650000 745.750000 3336.000000 744.000000 173052651 11 1.048884 0
$AACH_$AACH ERCH EGMX B752 223614 223641 77 90 0 BCS3318 131219 131219 3336.000000 744.000000 3336.016667 741.116667 173052651 12 1.629067 0
$AACH_$AACH ERCH EGMX B752 223641 223712 90 107 0 BCS3318 131219 131219 3336.016667 741.116667 3336.050000 736.300000 173052651 13 2.721423 0
$AACH_$AACH ERCH EGMX B752 223712 223745 107 120 0 BCS3318 131219 131219 3336.050000 736.300000 3336.083333 731.500000 173052651 14 2.711970 0
$AACH_$AACH ERCH EGMX B752 223745 223801 120 120 2 BCS3318 131219 131219 3336.083333 731.500000 3336.083333 729.566667 173052651 15 1.092231 0
$AACH_$AACH ERCH EGMX B752 223801 223814 120 124 0 BCS3318 131219 131219 3336.083333 729.566667 3336.100000 727.650000 173052651 16 1.082940 0
$AACH_$AACH ERCH EGMX B752 223814 223910 124 150 0 BCS3318 131219 131219 3336.100000 727.650000 3336.150000 718.983333 173052651 17 4.896375 0

```

Figura 65: exemple arxiu format .s06

Aquests arxius estan formats per una gran quantitat de dades, d'entre elles hi ha el nom, origen i destinació del vol, el tipus d'avió, l'estat, informació relativa als segments, matrícula, etc.

Aquest proveïdor també permet moltes facilitats de tractament de dades però en general sol presentar menys eficiència que *Flightstats*. Les principals deficiències són:

- Els arxius necessiten ser descarregats en format comprimit (prescindir d'una memòria).
- Existeix una presència d'informació redundant en els arxius.
- La informació obtinguda és recent, però no és possible obtenir-la a temps real.
- Només permet una quantitat limitada de descàrregues setmanals.

Malgrat aquests inconvenients, no deixa de ser un recurs útil i fiable per l'activitat de *Stratway*.

Una vegada es té el coneixement bàsic del funcionament d'aquesta eina i de les fonts principals amb que es treballaran, es procedirà a provar aquesta teoria a través de l'exemplificació d'un procés de resolució. D'aquesta manera es podrà entendre més a fons les funcions de les diferents classes existents.

```
super("AGENT Conflict/Collision Resolution");
//ImageIcon img = new ImageIcon("images/AGENT.png");
//setIconImage(img.getImage());

Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();

JToolBar toolbar = new JToolBar("");

JButton btnTrj = new JButton(new ImageIcon("images/Trj.png"));
btnTrj.setToolTipText("Load flight trajectories");
btnTrj.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        JOptionSource source_file = new JOptionSource();

source_file.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        source_file.setVisible(true);
        source_file.setLocationRelativeTo(null);
```

Aquest es un exemple de codi de creació del botó btnTrj, punt d'inici del exemple de seguiment, destinat a permetre l'usuari la càrrega de trajectòries. Com es pot observar s'utilitzen molts recursos visuals per la seva creació com la incorporació de una icona, la configuració de mida i de la posició en la pantalla. La classe *GUI_Stratway* està repleta d'elements que faciliten l'enteniment de la composició de la interfície.

Prenent el paper d'un usuari amb una informació de trajectòries diverses interessat en la resolució dels possibles conflictes.

El procés que segueix és senzill:

Parteix des de la interfície de *Stratway*, prement el botó `btnTrj`, on se li demana que introdueixi el arxiu, però abans ha de passar un filtre que li permetrà reduir el procés.

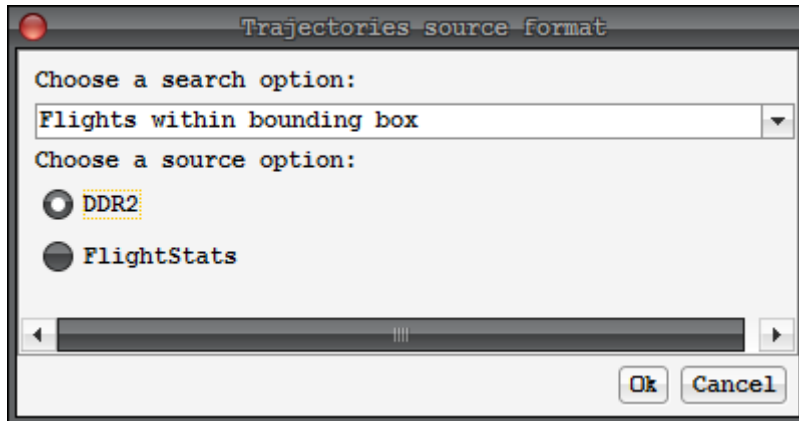


Figura 66: interfície gràfica de filtre intern

Aquest filtre intern de la interfície li permet escollir el tipus de cerca que vol i la font d'on proveiran els plans de vol. En el cas que esculli qualsevol tipus de cerca sobre la font *FlightStats*, el filtre ja esta pràcticament creat i és més directe ja que només cal introduir les dades que et demana la mateixa pàgina web: una *AppID* i una *AppKey* que t'identifiquin com a usuari registrat, i la informació necessària per efectuar el tipus de cerca.

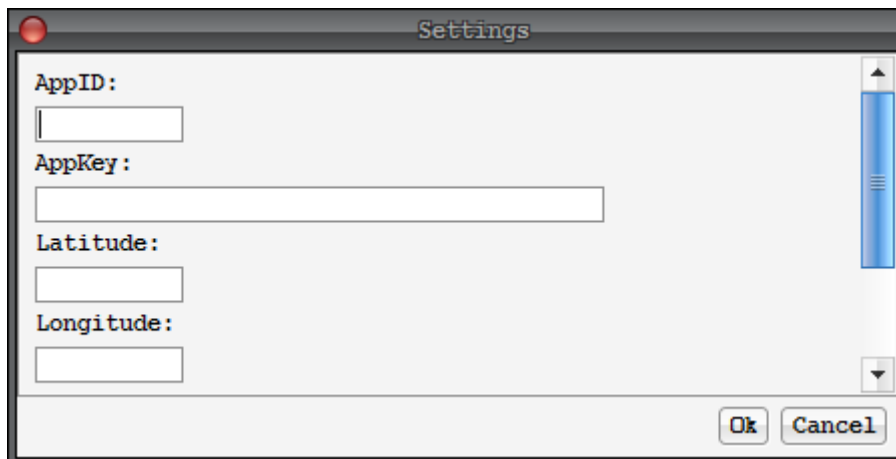


Figura 67: interfície gràfica de introducció de dades

En el cas de *DDR2*, s'ha de replicar el procés. *DDR2* com s'ha observat anteriorment, normalment ofereix arxius d'una gran quantitat d'informació. Per aquest motiu cal identificar totes les dades que es necessiten per tal d'aplicar

un filtre similar. Per tant la finestra no deixa de ser la mateixa, però en aquest cas cal crear les funcions necessàries (dintre de *Stratway*) per tal que es tinguin en compte els valors introduïts sobre l'arxiu descarregat.

4.3.2. Extracció de dades: *DataProcessor.java*

Una vegada s'ha escollit la font i la cerca desitjada, es crearà automàticament un arxiu. Aquest procés d'extracció de dades el realitza la classe *DataProcessor.java*, (amb el suport directe de les classes *RoutePoint.java* i *Trajectory.java*).

Aquesta classe ofereix la possibilitat de convertir les diferents dades en format *.csv*, és a dir, permet formatar les dades per tal que siguin compatibles amb l'eina. Com es pot veure, en el interior de la classe estan definits els dos convertidors: per DDR2 conversió de *.s06* (a través d'un arxiu comprimit *.7z*) a *.csv*, per *Flightstats* conversió de *.json* a *.csv*.

Les funcions que s'utilitzen en una extracció de dades a partir de la font *Flightstats* són les següents:

```
public static JSONObject readJsonFromUrl(String url) throws Exception;
private static String readAll(Reader rd) throws IOException;

public static void processDataFlightStats(String appId,String appKey,
double lat, double lon, int miles, int maxFlights) throws Exception;

private static int Conversor(String date);

private static void WriteCSV(FileWriter fileWriter,String[] line);
```

La primera funció *readJsonFromUrl*, s'encarrega de llegir un tipus d'arxiu JSON via enllaç URL, utilitzant la segona funció (*readAll*) per tractar i escanejar iterativament la informació que se'n obté. En la tercera funció anomenada *processDataFlightStats* amb la complementació de la quarta funció (*writeCSV*)

és on s'inicialitza el procés de transformació de la informació en un arxiu de text (.csv): primer es crea i s'anomena el arxiu, seguidament s'obté la informació que es necessita a través d'un objecte JSON i en funció dels valors dels paràmetres requerits (*lat*, *lon*, *miles*, *maxFlights*), es converteix aquest objecte en un format *String* i finalment es traspasa la informació seleccionada a un fitxer separat per comes.

Les funcions que s'utilitzen en una extracció de dades a partir de la font DDR2 són les següents:

```
public static List<Trajectory> readDDR2SO6(String filename);

private static void processDataDDR(List<Trajectory> listOfTrj);

public          static          void          writeToCSV(FileWriter
csvFileWriter,List<RoutePoint> routepoints);
```

El tractament de la font DDR2 és similar a l'anterior: en la funció *readDDRS06* s'introdueix el arxiu amb el format *s06* i es realitza un filtre de les dades necessàries per traspasar-ho al arxiu .csv, seguidament passa a la funció *processDataDDR* on es crea aquest arxiu .csv i mitjançant la funció *writetoCSV* es transcriu la informació en el fitxer objectiu.

```
private static final String COMMA_DELIMITER = ",";
private static final String new_line = "\n";
private static final String FILE_HEADER = "NAME, latitude, longitude,
altitude, time, aircraft_icao";
private static FileWriter fileWriter=null;
private static final int NUM_TRAJECTORIES = 7;
```

El arxiu que es pretén crear es constituirà d'una divisió amb comes entre els valors especificats (nom, latitud, longitud, altitud, temps, identificació) i una separació interlineal dels diferents *waypoints*. Es limitarà l'extracció a només 7 trajectòries ja que sinó es faria impossible obtenir una visualització dels resultats de la resolució.

4.3.3. Divisor de *waypoints*: *LegConversor.java*

En el moment que s'ha obtingut el arxiu filtrat segons els nostres interessos, es procedirà a demanar al usuari quin rang de separació entre *waypoints* vol que tinguin els seus vols.

Com es normal, tant *Flightstats* com DDR2 ofereixen unes trajectòries que no necessàriament han de tenir la mateixa separació de punts. En el cas de *Flightstats* ofereix un format estàndard de 60 segons de diferència entre *waypoint-waypoint*. En canvi DDR2 no segueix aquesta mateixa lògica.

Per això es creu que pot ser una bona idea crear una funcionalitat que permeti adequar aquest temps en funció a les necessitats. La classe que s'encarrega d'aquesta tasca s'anomena *LegConversor.java*, treballant juntament amb les classes *CsvReader.java* i *CsvWriter.java*.

```
public static void InterpolationBetweenPoints(FileWriter fw, String[]
first, String[] second, int desired);

private static void WriteCSV(FileWriter fileWriter,String[] line);

public static void FinalCSVWriter(String csvFile, int separation);
```

Les funcions de la classe que s'han utilitzat són les anteriors: *InterpolationBetweenPoints* serveix per aplicar els càlculs necessaris per obtenir (en funció del temps) els diferents *waypoints* corresponents a la nova *time-separation* sol·licitada. Les dues funcions restants s'encarreguen de llegir el fitxer origen i re-escriure'l amb les modificacions noves.

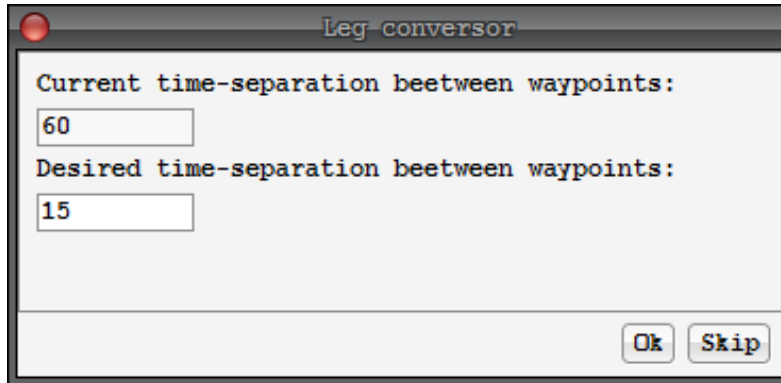


Figura 68: interfície gràfica de LegConversor.java

Aquesta classe permet personalitzar la separació en funció del temps dels diferents punts que componen una trajectòria. Això ho aconsegueix amb una simple interpolació. Per exemple: s'obtenen un total de tres plans de vol que ofereixen informació sobre els seus *NavPoints* cada 60 segons. Però es decideix que aquesta diferència de temps és massa gran per determinar correctament les possibles solucions dels conflictes entre les tres trajectòries i que seria necessari reduir el temps fins a 15 segons (per exemple). Doncs s'introdueix el número de segons desitjat i per mitjà d'una interpolació, aquesta classe crea noves dades estimades que permetin visualitzar una localització més exacte de l'aeronau.

NAME	latitude	longitude	altitude	time	aircraft_icao
WOW831	63.6734	-22.0365	8166	48499	736048664
WOW831	63.7168	-22.1141	7309	48559	736048664
WOW831	63.7581	-22.1883	6452	48619	736048664
WOW831	63.7973	-22.259	5594	48679	736048664
WOW831	63.8345	-22.3262	4737	48739	736048664
WOW831	63.8696	-22.3899	3880	48799	736048664
WOW831	63.9027	-22.4501	3023	48859	736048664
WOW831	63.9337	-22.5068	2165	48919	736048664
WOW831	63.9627	-22.5598	1308	48979	736048664
WOW831	63.9896	-22.6093	451	49039	736048664
ICE212	63.9716	-22.2602	2388	48522	736084112
ICE212	63.9637	-22.1528	2988	48582	736084112

Figura 69: comparació entre arxius .csv: arxiu anterior

NAME	latitude	longitude	altitude	time	aircraft_icao
WOW831	63.6734	-22.0365	8166	48499	736048664
WOW831	63.68425	-22.0559	7952	48514	736048664
WOW831	63.6951	-22.075300000000002	7738	48529	736048664
WOW831	63.705949999999994	-22.094700000000003	7524	48544	736048664
WOW831	63.7168	-22.1141	7309	48559	736048664
WOW831	63.7168	-22.1141	7309	48559	736048664
WOW831	63.727125	-22.13265	7095	48574	736048664
WOW831	63.73745	-22.151200000000003	6881	48589	736048664
WOW831	63.747775000000004	-22.169750000000004	6667	48604	736048664
WOW831	63.7581	-22.1883	6452	48619	736048664
WOW831	63.7581	-22.1883	6452	48619	736048664
WOW831	63.7679	-22.205975000000002	6238	48634	736048664
WOW831	63.777699999999996	-22.223650000000003	6024	48649	736048664
WOW831	63.787499999999994	-22.241325000000003	5810	48664	736048664
WOW831	63.7973	-22.259	5594	48679	736048664

Figura 70: comparació entre arxius .csv: arxiu posterior

La primera captura il·lustra les dades referents al arxiu inicial extret gràcies a la classe *DataProcessor.java* en el que es llisten els diferents *waypoints* amb una diferència temporal de 60 segons(*JSONconvert2.csv*). En la segona captura es mostra la divisió dels *waypoints* corresponents a l'arxiu inicialitzada amb la classe *legConversor.java*, en aquest cas, amb una reducció de 15 segons afegint 3 *waypoints* extres per cada separació existent (*LegConversion.csv*).

Arribat a aquest punt, s'ha obtingut un arxiu de dades en el format indicat, en base al tipus de cerca desitjada, amb origen de la font escollida i a més a més, amb una distribució o divisió dels *waypoints* personalitzada. Si no han sorgit problemes, el següent pas és buscar aquest arxiu correctament desat en una carpeta per defecte, i carregar-lo a *Stratway*.

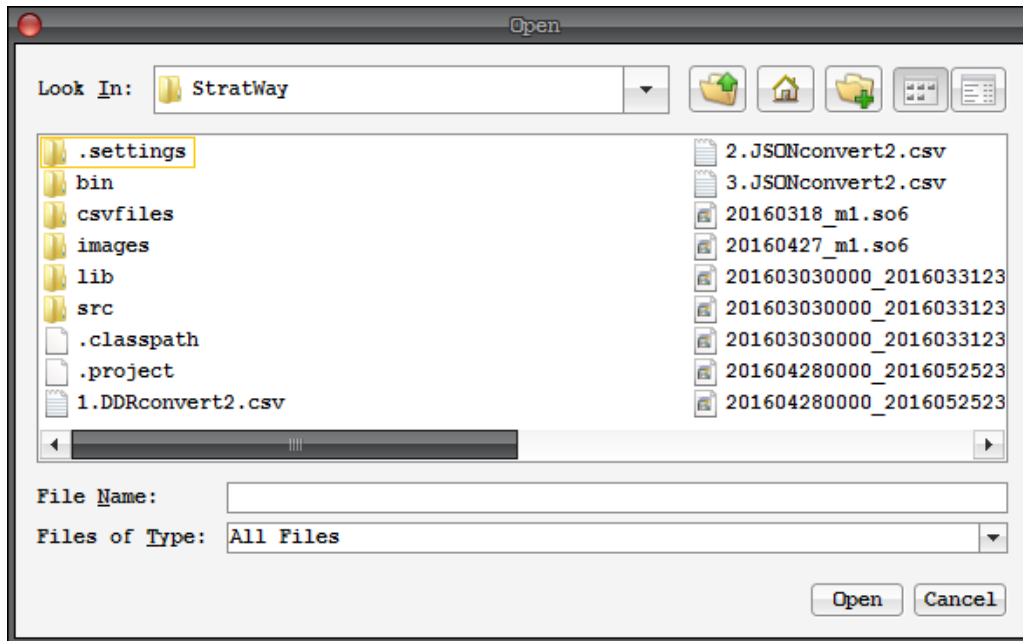


Figura 71: Finestra de càrrega d'arxius

Un cop carregat el nou arxiu, es mostra la informació continguda d'una forma visual a través de les sis diferents perspectives, conjuntament amb una descripció textual al panell inferior. En aquest pas es pot començar a analitzar el tipus d'escenari que es troba, i es podria afegir nous condicionants com el l'afectació de l'estat climatològic d'aquella regió, però aquesta funcionalitat encara no ha estat desenvolupada.

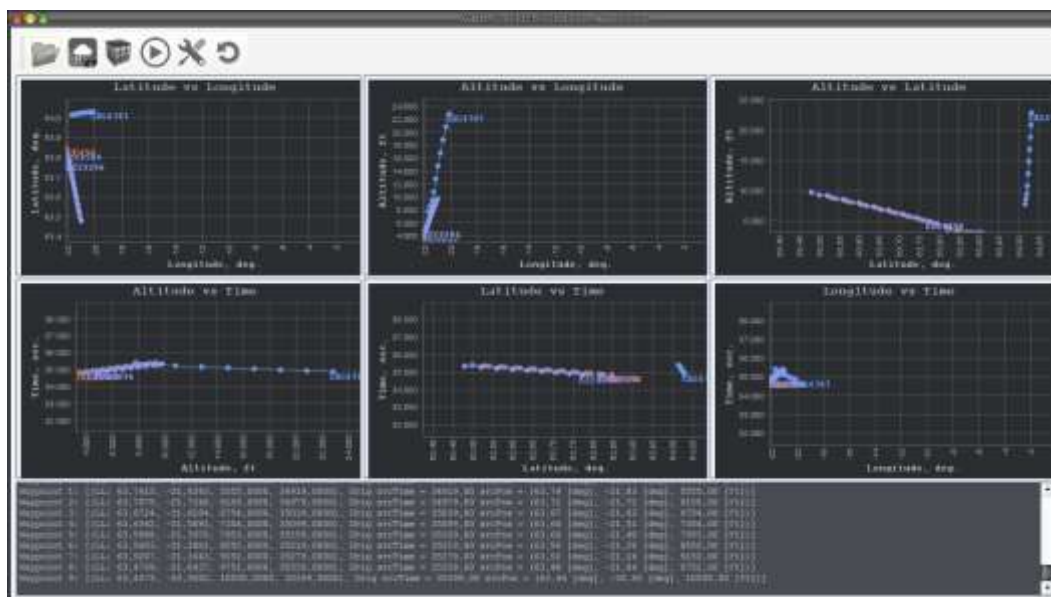


Figura 72: Captura de pantalla amb trajectòries GUI_Stratway

4.3.4. IterativeCDCR.java (future work)

Per tant, l'últim pas és el de resoldre automàticament els possibles conflictes prement el botó *Solve* que ofereix la interfície. Per mirar d'entendre aquesta part ens hem de fixar en el procés de resolució prèviament explicat i analitzar les diferents estratègies que hi aplica.

```
private static void runCDCR() {
    _solution = _sw.resolveConflicts();
    // See the resolution
    if ( !_solution.equals(_own) ) {
        try{
            _doc.insertString(_doc.getLength(), "New flight plan is:\n", _keyWord );
            _doc.insertString(_doc.getLength(), "\n" + _solution, _keyWord );
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        String result = "";
        // Check if conflict(s) were resolved
        if ( _sw.getResolutionStatus() ==
Stratway.ResolutionStatusValue.CONFLICT_FREE ) {
            result = "All conflicts resolved.";
        } else {
            result = "Not all conflicts resolved.";
        }
        try
        {
            _doc.insertString(_doc.getLength(), result+"\n", _keyWord );
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    else {
        try{
            _doc.insertString(_doc.getLength(), _sw.toString() + "\n", null);
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    if ( _sw.hasError() ) {
        System.out.println(_sw.getMessage());
    }
}
```

A través d'aquesta funció anomenada *RunCDCR()* s'executa el procés de resolució dels conflictes existents (o no existents) de les trajectòries introduïdes al programa. Aquest procés és molt similar a l'exemple exposat en la introducció de l'eina *Stratway*: primer s'aplica la resolució i s'analitzen els efectes, seguidament es comportava si la trajectòria inicial ha canviat (fet que significaria que s'han resolt els conflictes existents) i en cas contrari, significa que no existeixen conflictes entre el pla *ownship* i la resta de tràfics. En cas que es detectin canvis, es crea un nou pla de vol amb els nous *waypoints*. Finalment es comprova el estat de resolució resultant per validar que tots els conflictes hagin estat resolts o si han sorgits errors, que es comuniqui el missatge.

```
All strategies failed at time 34831,00
----- Time 34873,00 -----
SideStep cannot be used if conflict is on current leg.
SideStep did not find a solution
TrackEarly did not find a solution
Vertical did not find a solution*
Speed did not find a solution
Track did not find a solution
All strategies failed to resolve the conflictName, Lat, L
```

Figura 73: Panell d'informació sense solucions

En aquesta figura s'hi representa el missatge en el panel de sortida on s'hi observa com la resolució no ha estat exitosa i se'ns mostra en quin temps ha fallat l'algorisme de resolució, quines han estat les estratègies que no han trobat una solució i quin estat de resolució és el resultant.

Per entendre aquest missatge és de gran importància entendre la lògica d'aplicació de les estratègies i quins són els paràmetres en els que es basa, ja que aquesta és la funció que dóna sentit a *Stratway*.

En la següent llista es mostren alguns dels paràmetres principals que permeten la definició dels límits d'aplicació de les estratègies. Com es pot apreciar alguns d'ells van dirigits a definir els límits màxims i mínims de velocitat

vertical como horizontal, d'acceleració o d'altitud. Altres variables emmagatzemen el temps necessari per efectuar la detecció i la resolució, etc.

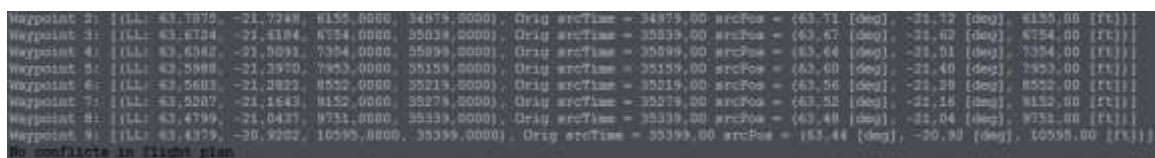
Tots aquestes variables reservades (i més) van destinades a poder configurar els condicionants dels que depenen les possibles solucions que se'n puguin extreure.

```
double D          = Units.from("nm", 5.0); // protected distance,
horizontal (dcore)
double H          = Units.from("ft", 950.0); // protected distance,
vertical (dcore)
double D_det_buf  = Units.from("nm", 0.0); // protected distance,
horizontal (rcore)
double H_det_buf  = Units.from("ft", 0.0); // protected distance,
vertical (rcore)
double D_res_buf  = Units.from("nm", 0.2); // protected distance,
horizontal (rcore)
double H_res_buf  = Units.from("ft", 150.0); // protected distance,
vertical (rcore)
double T_d        = Units.from("second", 100.0); // lookahead time for
detection (used in dcore)
double T_r        = Units.from("second", 100.0); // lookahead time for
resolution (used in dcore)
double T_p        = Units.from("second", 100.0); // lookahead time for
polygon detector
double minBandsLookaheadTime = Units.from("min", 3.0); // minimum
lookahead time for bands calculations
double bandsLookaheadTime = -1.0; // minimum lookahead time for bands
calculations
double minGroundSpeed = Units.from("kn", 50.0); // lower bound on the
ground speed that a strategy will allow
double maxGroundSpeed = Units.from("kn", 700.0); // upper bound on the
ground speed that a strategy will allow
double maxBankAngle = Units.from("deg", 30.0); // acceptable bank
angle for kinematic bands
double maxGsAccel = 2; // acceptable ground speed acceleration
double maxVsAccel = 1; // acceptable vertical speed acceleration
double maxVerticalSpeed = Units.from("fpm", 5000.0); // upper bound on
the vertical speed that a strategy will allow
double minVerticalSpeed = Units.from("fpm", -5000.0); // lower bound on
the vertical speed that a strategy will allow
double maxAltitude = Units.from("ft", 50000.0); // upper bound on the
altitude that a strategy will allow the resolution to use
double minAltitude = Units.from("ft", 0.0); // lower bound on the
altitude that a strategy will allow the resolution to use
double latLonAccuracy = Units.from("nm", 0.10); // this determines how
far apart virtual waypoints are inserted
```

```
double isolationFactor = 2.0; // governs the size of the isolation
region
boolean redundantLoSCheck = false; // determines whether an extra
great-circle level LoS check is performed.
double pilotApprovalTime = 5.0; // sets a minimum time from the
timeOfCurrentPosition to when a maneuver can begin
```

Per trobar les trajectòries més òptimes considerant els paràmetres exposats i els conflictes existents, *Stratway* utilitza quatre estratègies que es defineixen en les següents classes: *StrategyGsMeister.java*, *StrategySideMeister.java*, *StrategyTrackMeister.java*, *StrategyVerticalMeister.java*. Aquestes estratègies actualment s'executen automàticament seguint un ordre predeterminat.

En aquest punt també el futur algorisme s'encarregarà de, a través de la utilització de les classes de les quatre estratègies i dels paràmetres, realitzar una iteració d'aquesta funció prèviament exposada (*RunCDCR()*) alternant l'assignació del pla subjecte (*ownship*) al llarg dels diferents tràfics existents. La classe que aportarà aquesta funcionalitat s'anomena *IterativeCDCR.java*, tot i que actualment no es pugui apreciar la seva incorporació. D'aquesta manera es comprovarà les diferents maneres de resoldre els conflictes des de diferents punts de partida.



```
Waypoint 2: [LL: 43.1875 -21.7248 8155.0000 34979.0000] Orig arrTime = 34979.00 arrPos = (43.71 [deg], -21.72 [deg], 8155.00 [ft])
Waypoint 3: [LL: 43.6794 -21.6184 8734.0000 35039.0000] Orig arrTime = 35039.00 arrPos = (43.67 [deg], -21.60 [deg], 8734.00 [ft])
Waypoint 4: [LL: 43.6382 -21.5091 7354.0000 35099.0000] Orig arrTime = 35099.00 arrPos = (43.64 [deg], -21.51 [deg], 7354.00 [ft])
Waypoint 5: [LL: 43.5988 -21.3970 7953.0000 35159.0000] Orig arrTime = 35159.00 arrPos = (43.60 [deg], -21.40 [deg], 7953.00 [ft])
Waypoint 6: [LL: 43.5603 -21.2822 8552.0000 35219.0000] Orig arrTime = 35219.00 arrPos = (43.56 [deg], -21.39 [deg], 8552.00 [ft])
Waypoint 7: [LL: 43.5207 -21.1643 9152.0000 35279.0000] Orig arrTime = 35279.00 arrPos = (43.52 [deg], -21.16 [deg], 9152.00 [ft])
Waypoint 8: [LL: 43.4799 -21.0437 9751.0000 35339.0000] Orig arrTime = 35339.00 arrPos = (43.48 [deg], -21.04 [deg], 9751.00 [ft])
Waypoint 9: [LL: 43.4379 -20.9252 10350.0000 35399.0000] Orig arrTime = 35399.00 arrPos = (43.44 [deg], -20.92 [deg], 10350.00 [ft])
No conflicts in flight plan
```

Figura 74: Panell d'informació amb resolució

Finalment, en funció de si existeixen conflictes entre els tràfics originals i de si l'eina els ha pogut resoldre correctament, s'obté un resultat en el qual s'hi representarà els nous plans de vol amb una resolució dels conflictes exitosa.

5. CONCLUSIONS

5.1. Resultats obtinguts

Al llarg del desenvolupament d'aquest projecte s'ha pogut comprovar que realment les previsions del trànsit aeri indiquen que existeix i seguirà existint un augment exponencial durant els propers anys. Aquest gran creixement demana constantment noves idees i propostes per tal de solucionar tots els inconvenients que puguin aparèixer durant el procés, amb la finalitat d'aconseguir satisfer aquesta demanda de capacitat, eficiència i seguretat.

La elaboració d'aquesta tesis ha estat sobretot dirigida a analitzar i desenvolupar un ambient de proves per provar els conceptes de la propera generació d'aquest demandat nou sistema ATM europeu. Aquest objectiu s'ha descompost en diverses fases que han permès arribar a aquest punt final de la tesis: començant per un anàlisi i descripció general de la situació real del marc d'actuació, seguit d'una explicació dels conceptes que envolten el projecte sobre el que s'ha treballat, una presentació de l'ambient de proves en el que es participarà començant amb una proposta de disseny d'arquitectura i, finalitzant el document, amb una mostra de treball realitzat sobre una de les eines principals.

S'ha justificat la idea de que el sector aeronàutic, concretament en la gestió del tràfic aeri, està patint canvis i s'està actuant activament per afrontar-los, considerant opcions que no només tinguin en compte el progrés sinó també l'estabilitat i respecte per l'entorn, on projectes com SESAR, signifiquen una acció cooperativa i internacional amb la intenció de trobar les millors solucions davant de diferents problemes comuns.

També s'ha demostrat que tècniques innovadores com l'aplicació de nous sistemes com la comunicació multi-agent o un perfeccionament del sistema encarregat de la detecció de possibles col·lisions, poden contribuir a millorar molts dels processos que es porten a terme en l'actualitat.

En la fase de l'anàlisi de l'aplicació, s'ha pogut experimentar com és important tenir una prèvia idea de la que sorgeixi una primera imatge de l'aplicació. La definició dels requisits a complir i les interrelacions entre aquests, ha estat una manera molt fàcilment descriptiva de les característiques figuratives i funcionals de l'aplicació, que ha permès establir unes bases molt clares de les que parteixen la definició més gràfica de l'estructura.

En la fase de la part de disseny, inici de la part pràctica del projecte, s'ha aconseguit donar una primera idea conceptual de l'arquitectura i la organització de l'aplicació. En aquest punt se'n pot destacar el grau de dificultat i d'exigència que demana una representació que busca plasmar el màxim de claredat. En el projecte es pot observar com s'han il·lustrat diferents vistes de l'arquitectura i com aquesta ha anat evolucionant a mesura que s'han concretat alguns aspectes que primerament quedaven molt difusos. Gràcies a diferents representacions esquemàtiques han facilitat mostrar quin és el funcionament esperat de l'aplicació. Tampoc s'ha d'oblidar que només representa una proposta de disseny i que no necessàriament ha de significar la única i més adequada possibilitat, moltes de les condicions que s'han definit probablement siguin modificades en la futura fase d'implementació.

En l'apartat del cas d'estudi i de creació de codi, s'han completat la major part de reptes establerts: s'ha obtingut del funcionament i de les característiques de l'eina *Stratway*, s'ha afegit una part addicional de codi font per tal d'incrementar el seu ventall de funcionalitats i s'ha exposat un exemple que defineix d'una forma precisa el procés que es segueix. També és

important esmentar que han existit inconvenients i des ajustaments de les expectatives. Algunes de les funcions han de ser revisades, comprovades i corregides per tal de perfeccionar-ne el seu paper, ja que algunes d'elles no contemplen en plenitud el funcionament esperat, per exemple, la funció `LegConversor.java` no té en compte la variació de la velocitat en el temps, entre d'altres. També cal considerar que existeixen múltiples funcions ideades que encara no s'han implementat en el funcionament, però es preveu fer-ho en un futur treball.

Per tant, arribat en aquest punt final, no es pot concloure amb un resultat concret que representi el fruit d'aquests mesos invertits en l'elaboració del treball, degut a que no es partia d'una hipòtesi concreta, però si que es pot afirmar el compliment dels objectius que es fixaren en funció d'unes tasques pendents a realitzar, permetent un marge de llibertat en la seva realització. Malgrat això, també cal destacar una ampliació en comprensió de conceptes relatius a l'ATM actual i dels reptes que aquest planteja, i com l'aplicació de diferents tècniques porten a la proposta de noves maneres d'afrontar-los.

5.2. Futur treball

Considerant que aquest projecte s'inicià davant d'unes necessitats reals, suportat com a projecte realitzat en empresa i focalitzat en l'estudi de la fase inicial del seu conjunt, es fa evident que existeixi una futura prolongació del treball pendent a realitzar.

Existeix una planificació de successos del projecte AGENT on es descriuen una sèrie de passos per tal que permetin completar, en el termini desitjat, totes les parts del projecte. En aquesta tesis s'hi avarca la fase d'inicialització, que comprèn les tasques de recerca d'informació útil i necessària, i l'anàlisi i disseny de l'aplicació. Per tant és obvi que queda un gran període de treball,

on es perfeccionaran moltes de les propostes plantejades i s'implementaran aquestes idees conceptuals a través d'un desenvolupament pràctic.

Una visió personal del futur treball correspondria a l'afirmació de que es tracta d'un projecte amb un abast enorme. La seva completa construcció i validació representarà una gran aportació per a la gestió del tràfic aeri, però per aconseguir aquesta fita també és cert que el procés no serà immediat ni fàcil. Queden moltes qüestions per resoldre que encara no es poden estudiar degut a la desconexença viable sobre molts temes, generada a causa de la novetat de la proposta. Però d'aquesta sensació se'n pot treure una deducció positiva: si es disposés d'una seguretat i garantia en totes les coses, perdria sentit el tret distintiu que caracteritza un projecte innovador, que és el d'aportar idees prèviament mai ideades i intentar potenciar la cooperació laboral per a la seva materialització, i al cap d'avall, per al impacte positiu que comportarà.

6. REFERÈNCIES BIBLIOGRÀFIQUES

6.1. Enllaços web:

1. Introducció a conceptes de l'ATM actual.
<http://www.icao.int/safety/airnavigation/pages/atm.aspx>
2. Principals serveis i objectius del sistema ATM.
<https://www.qinetiq.com/services-products/c4isr/Pages/air-traffic-management.aspx>
3. Visió general del concepte del cel únic europeu.
[http://www.skybrary.aero/index.php/Single_European_Sky_\(SES\)](http://www.skybrary.aero/index.php/Single_European_Sky_(SES))
4. Breu explicació del programa SESAR.
<http://www.enaire.es/csee/Satellite/navegacion-aerea/es/Page/1237551223699/>
5. Introducció al recent i millorat sistema de detecció de conflictes TCAS 7.1.
http://www.faa.gov/documentLibrary/media/Advisory_Circular/TCAS%20II%20V7.1%20Intro%20booklet.pdf
6. Introducció al concepte multi-agent
<http://coltech.vnu.edu.vn/http/media/courses/AI++/Tai%20lieu/TLTK.pdf>
7. Fonaments del disseny i desenvolupament de software
http://es.slideshare.net/jose_rob/diseo-de-la-arquitectura-del-software
<http://indalog.ual.es/mtorres/LP/FundamentosDiseno.pdf>
8. Introducció al sistema SWIM
<http://www.eurocontrol.int/swim>

9. Coneixement usos de diagrames UML

<http://www.uml-diagrams.org/>

6.2. Bibliografia documental:

1. Arxius PDF: **Class_2.pdf, Class_3.pdf, Class_4.pdf, Class_5.pdf, Class_10.pdf, Class_11.pdf**, Material docent corresponent a *Airport Operations*.

1. Arxius PWP: **Sesion1_reduced, Sesion2_reduced, Sesion3_2015, Sesion4_2015**, Material docent corresponent a *tècniques de navegació i control aeri*.

3. Arxius PDF: **Teoria 02 - Agents intel·ligents, Teoria 10 - Sistemes Multi-agent i IA distribuïda - Complet**, Material docent corresponent a *Intel·ligència Artificial*.

4. Arxius PWP: **AGENT_KoM_WP2_Feb2016, AGENT_KoM_WP3_Feb2016, AGENT_KoM_WP4_Feb2016, AGENT-Project-Summary, KICK_OFF_ASLOGIC_PRESENTATION_AGENT_TEMPLATE**, Presentacions relatives a la definició i l'evolució del desenvolupament del projecte AGENT.

5. Arxius PDF: **ddr2_userguide_generic, AGENT_GRAPHICAL_GUIDELINES_MANUAL, jt1de1, Part B_AGENT_FINAL, Meeting-Minutes-Agent, Table of attendees, h2020-mga-gga-multi_en, stratway-api**. Documentació recollida prèviament relativa al projecte AGENT i multi-agent.

6. Arxiu PDF: **01_thesis_Prologue_and_Chapter_1.pdf**, Introduction and chapter 1 of the thesis *STREAM project*, by Sergio Ruiz Navarro.

7. Arxiu PDF: **The_Effect_of_TBO_on_Air_Traffic_Complexity_2014-11-03**. Tesi doctoral, Tomislav Radisic.

8. Arxiu PDF: **Mukherjee_abstract**, Introducció de la tesi "ROLE OF OPTIMIZATION MODELS IN FUTURE AIR TRAFFIC MANAGEMENT SYSTEM" i presentació de conceptes principals.
9. Arxiu PWP: **Exploratory Research Project Presentation template**, Presentació del propòsit general del projecte de recerca i investigació.

6.3. Bibliografia escrita:

1. Bertsimas, D. & Patterson, S. S. (2000). *The traffic flow management rerouting problem in air traffic control: a dynamic network flow approach*, *Transportation Science* 34(3): 239- 255.
2. Nicolas Barnier, Pascal Brisset & Thomas Riviere (2001): *Slot allocation with constraint programming: Models and results*. In: Hugh McLaurin, editor: *Proceedings of ATM'01, the 4th USA/Europe R&D Seminar on Air Traffic Management*, Santa Fe, NM, USA.
3. *European Organisation for the safety Air Navigation EUROCONTROL*, *Manual del usuari referent a la informació de l'avió (BADA) REVISION 3.10*, Abril 2012.
4. V. Gorodetsky, O. Karsaev, V. Samoylov, and V. Skormin. *Multi-agent Technology for Air Traffic Control and Incident management in airport airspace*. In *proceedings of the International workshop agents in traffic and transportation*, pages 119-125. *International Foundation for Autonomous Agents and Multiagent Systems*, 2008.

7. ANNEXES I APÈNDIXS

(Complets en format digital)

Signatura a l'últim full

(annexos nomes amb el cd)